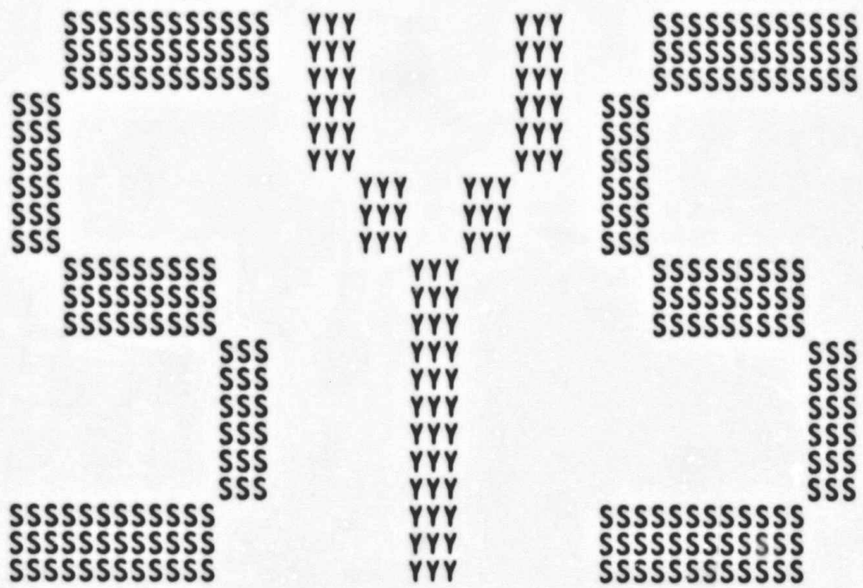
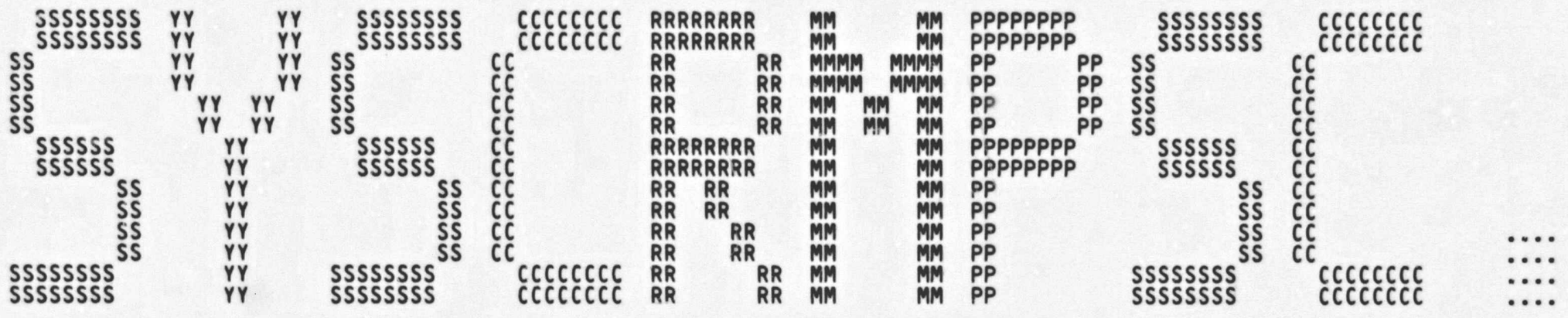

SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS


SYSCROMPSC

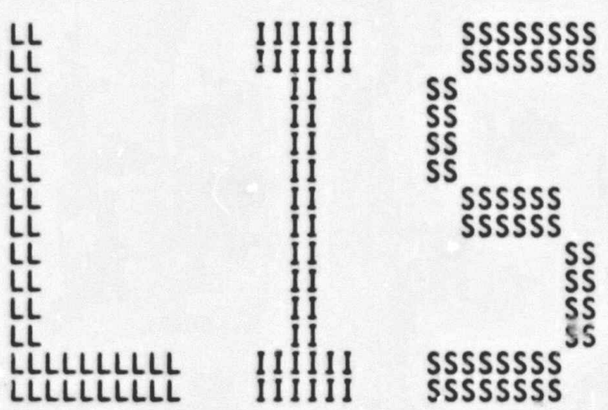

LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

Table of contents

(2) 159 DECLARATIONS
(3) 221 CRMPSC - CREATE AND MAP SECTION
(4) 828 MGBLSC - MAP GLOBAL SECTION
(6) 1409 FAST_MAP - DO COMMON CASES EFFICIENTLY
(7) 1516 SETSECPROTOWN - SET SECTION PROTECTION AND OWNER
(8) 1589 INITSECTBL - ALLOC & INIT SECTION TABLE ENTRY
(9) 1758 MAP PROCESS SECTION
(10) 1901 MAPSECPAG - MAP A SINGLE PROCESS/GLOBAL SECTION PAGE
(11) 2081 CHECK_WINDOW INSURE FULLY MAPPED FILE
(11) 2214 MMG$RET_BYT_QUOTA RETURN BYTCNT QUOTA
(11) 2277 READ_RESIDENT - INITIALIZE A RESIDENT GLOBAL SECTION

```
0000     1          .TITLE  SYSCRMPSC - Create and Map Section System Service
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26
0000    27 ;++
0000    28 ; FACILITY:
0000    29 ;
0000    30 ; ABSTRACT:
0000    31 ;
0000    32 ; ENVIRONMENT:
0000    33 ;
0000    34 ; AUTHOR: PETER H. LIPMAN     .   , CREATION DATE: 24-APR-78
0000    35 ;
0000    36 ; MODIFIED BY:
0000    37 ;
0000    38 ;       V03-034 WMC0012         Wayne Cardoza           03-Jul-1984
0000    39 ;               Fix register problem with WMC0011.
0000    40 ;
0000    41 ;       V03-033 WMC0011         Wayne Cardoza           25-May-1984
0000    42 ;               Fix protection checks for CRF global sections.
0000    43 ;
0000    44 ;       V03-032 LMP0221         L. Mark Pilant,         12-Apr-1984  16:22
0000    45 ;               Modify the protection check to call EXE$CHECKPROT_16
0000    46 ;               instead of the various EXE$CHKxxxACCES routines.
0000    47 ;
0000    48 ;       V03-031 WMC0012         Wayne Cardoza           9-Apr-1984
0000    49 ;               Fix check for resident section read complete.
0000    50 ;
0000    51 ;       V03-030 CDS0001         Christian D. Saether    9-Apr-1984
0000    52 ;               Credit BYTLM as well as BYTCNT when returning quota
0000    53 ;               as a result of creating shared WCB.  It is now being
0000    54 ;               charged by the file system.
0000    55 ;
0000    56 ;       V03-029 WMC0011         Wayne Cardoza           28-Mar-1984
0000    57 ;               Check that page isn't in memory seen by MP before
```

```
0000     58 ;                bumping the inhibit count.
0000     59 ;                Add hash byte to GSD.
0000     60 ;
0000     61 ;        V03-028 WMC0010         Wayne Cardoza           14-Mar-1984
0000     62 ;                Set valid bit in GPTE for resident global.
0000     63 ;
0000     64 ;        V03-027 WMC0009         Wayne Cardoza           22-Feb-1984
0000     65 ;                Add resident global section support.
0000     66 ;
0000     67 ;        V03-026 MSH0005         Michael S. Harvey       3-Feb-1984
0000     68 ;                R5 is now a valid input for access checking routines. Make
0000     69 ;                sure R5 is correct before calling them.
0000     70 ;
0000     71 ;        V03-025 MSH0004         Michael S. Harvey       26-Jan-1984
0000     72 ;                Add support for lengthened global section names in GSDs.
0000     73 ;
0000     74 ;        V03-024 WMC0008         Wayne Cardoza           12-Jan-1983
0000     75 ;                Process sections should not use fast path for PFN sections.
0000     76 ;
0000     77 ;        V03-023 TMK0001         Todd M. Katz            11-Dec-1983
0000     78 ;                Fix a broken branch.
0000     79 ;
0000     80 ;        V03-022 WMC0007         Wayne Cardoza           23-Sep-1983
0000     81 ;                Changes to routines to fill in PTEs to improve performance.
0000     82 ;
0000     83 ;        V03-021 CWH3021         CW Hobbs                18-Sep-1983
0000     84 ;                Fix broken branch.
0000     85 ;
0000     86 ;        V03-020 WMC0006         Wayne Cardoza           30-Aug-1983
0000     87 ;                Disable IPL 0 wait on process section creation.
0000     88 ;
0000     89 ;        V03-019 ADE9005         Alan D. Eldridge        31-May-1983
0000     90 ;                Change BSBW to JSB's to MMG$RETADRINI and MMG$INADRINI.
0000     91 ;
0000     92 ;        V03-018 RSH0031         R. Scott Hanna          26-May-1983
0000     93 ;                Fix linker truncation errors.
0000     94 ;
0000     95 ;        V03-017 WMC0005         Wayne Cardoza           24-May-1983
0000     96 ;                Allow mapping of execute-access global sections.
0000     97 ;
0000     98 ;        V03-016 LJK0199         LAWRENCE J. KENAH       12-Apr-1983
0000     99 ;                Restore FILCNT along with BYTCNT for files with shared windows.
0000    100 ;
0000    101 ;        V03-015 WMC0004         Wayne Cardoza           23-Mar-1983
0000    102 ;                Fix a linker truncation error.
0000    103 ;
0000    104 ;        V03-014 WMC0003         Wayne Cardoza           18-Mar-1983
0000    105 ;                Fix a return status in mapsecpag.
0000    106 ;
0000    107 ;        V03-013 WMC0002         Wayne Cardoza           02-Mar-1983
0000    108 ;                MMG$CRECOM2 is gone
0000    109 ;                MMG$RETADRINI, MMG$INADRINI return status
0000    110 ;
0000    111 ;        V03-012 WMC53991        Wayne Cardoza           21-Feb-1983
0000    112 ;                Make protection checks on global sections not mapped to file.
0000    113 ;
0000    114 ;        V03-011 SRB0066         Steve Beckhardt         18-Feb-1983
```

```
0000    115 ;                   Fixed two broken branch displacments.
0000    116 ;
0000    117 ;       V03-010 WMC0001         Wayne Cardoza           02-Feb-1983
0000    118 ;                   Ensure that section table entry is complete before service
0000    119 ;                   touches the return address.
0000    120 ;
0000    121 ;       V03-009 KDM0037         Kathleen D. Morse       16-Dec-1982
0000    122 ;                   Prevent shared memory and PFNMAP sections from being
0000    123 ;                   backed to page file via SEC$V_PAGFIL.  Optimize some code
0000    124 ;                   and add some comments.
0000    125 ;
0000    126 ;       V03-008 KDM0036         Kathleen D. Morse       15-Dec-1982
0000    127 ;                   Re-compute the section table entry address after every
0000    128 ;                   page creation as the process header may have been expanded
0000    129 ;                   causing the section table to slide up one page.
0000    130 ;
0000    131 ;       V03-007 KDM0038         Kathleen D. Morse       21-Dec-1982
0000    132 ;                   Change BSBW to JSB.
0000    133 ;
0000    134 ;       V03-006 KDM0029         Kathleen D. Morse       10-Nov-1982
0000    135 ;                   Correct check for backwards mapping of shared memory
0000    136 ;                   global section.  Only return SS$_IVSSRQ if the section
0000    137 ;                   is mapped in pieces, not if it is mapped in one contiguous
0000    138 ;                   piece of memory.
0000    139 ;
0000    140 ;       V03-005 LJK0189         Lawrence J. Kenah       10-Nov-1982
0000    141 ;                   Make RELPAG parameter work correctly for shared memory
0000    142 ;                   global sections.
0000    143 ;       V03-004 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    144 ;                   Added $IODEF.
0000    145 ;
0000    146 ;       V03-003 PHL0101         Peter H. Lipman         21-Jun-1982
0000    147 ;                   $QIOW now synchronizes the EFN and IOSB parameters
0000    148 ;                   correctly.  Eliminate the synchronization code here.
0000    149 ;
0000    150 ;
0000    151 ;       V03-002 DWT0041         David Thiel             10-Jun-1982
0000    152 ;                   Correct sequence of clearing shared memory bitmap lock
0000    153 ;                   bit and releasing the shared memory GSD mutex so that the
0000    154 ;                   bitmap lock bit is cleared first.
0000    155 ;
0000    156 ;
0000    157 ;--
```

```
                     0000    159          .SBTTL  DECLARATIONS
                     0000    160 ;
                     0000    161 ; INCLUDE FILES:
                     0000    162 ;
                     0000    163          $ARMDEF                         ;ACCESS BITMASK DEFINITIONS
                     0000    164          $ARBDEF                         ;AGENT'S RIGHTS BLOCK OFFSETS
                     0000    165          $CCBDEF                         ;CHANNEL CONTROL BLOCK DEFINITIONS
                     0000    166          $DEVDEF                         ;DEFINE DEVICE CHARACTERISTICS
                     0000    167          $DYNDEF                         ;DYNAMIC DATA STRUCTURE DEFINTIONS
                     0000    168          $FCBDEF                         ;FILE CONTROL BLOCK DEFINITIONS
                     0000    169          $FIBDEF                         ;FILE INFORMATION BLOCK DEFINITIONS
                     0000    170          $GSDDEF                         ;GLOBAL SECTION DESCRIPTOR DEFINTIONS
                     0000    171          $IODEF                          ;I/O FUNCTION CODE DEFINITIONS
                     0000    172          $IPLDEF                         ;PROCESSOR PRIORITY LEVELS
                     0000    173          $IRPDEF                         ;I/O REQUEST PACKET
                     0000    174          $JIBDEF                         ;JOB INFORMATION BLOCK OFFSETS
                     0000    175          $MMGDEF                         ;OFFSETS FROM FP INTO SCRATCH AREA
                     0000    176          $ORBDEF                         ;OBJECT'S RIGHTS BLOCK OFFSETS
                     0000    177          $PCBDEF                         ;PROCESS CONTROL BLOCK DEFINITIONS
                     0000    178          $PFNDEF                         ;PFN DATA BASE
                     0000    179          $PHDDEF                         ;PROCESS HEADER DEFINITIONS
                     0000    180          $PRDEF                          ;PROCESSOR REGISTER DEFINTIONS
                     0000    181          $PRTDEF                         ;PROTECTION CODE DEFINITIONS
                     0000    182          $PRVDEF                         ;PRIVILEGE BIT DEFINITIONS
                     0000    183          $PSLDEF                         ;DEFINE PROCESS MODES
                     0000    184          $PTEDEF                         ;PAGE TABLE ENTRY DEFINITIONS
                     0000    185          $RSNDEF                         ;RESOURCE NAMES
                     0000    186          $SECDEF                         ;SECTION TABLE DEFINITIONS
                     0000    187          $SHBDEF                         ;SHARED MEMORY CONTROL BLOCK DEFINITIONS
                     0000    188          $SHDDEF                         ;SHARED MEMORY COMMON DATA PAGE DEF.
                     0000    189          $SSDEF                          ;SYSTEM STATUS CODE DEFINTIONS
                     0000    190          $UCBDEF                         ;DEFINE UNIT CONTROL BLOCKS
                     0000    191          $VADEF                          ;VIRTUAL ADDRESS VIELDS
                     0000    192          $WCBDEF                         ;WINDOW CONTROL BLOCK DEFINITIONS
                     0000    193 ;
                     0000    194 ; EXTERNAL SYMBOLS:
                     0000    195 ;
                     0000    196 ;
                     0000    197 ; MACROS:
                     0000    198 ;
                     0000    199 ;
                     0000    200 ; EQUATED SYMBOLS:
                     0000    201 ;
                     0000    202 ; OFFSET FROM AP
                     0000    203 ;
00000004             0000    204          INADR           = 4             ;OFFSET TO INPUT RANGE
00000008             0000    205          RETADR          = 8             ;OFFSET TO RETURN RANGE
0000000C             0000    206          ACMODE          = 12            ;ACCESS MODE
00000010             0000    207          FLAGS           = 16            ;MAP SECTION FLAGS
00000014             0000    208          GSDNAM          = 20            ;GLOBAL SECTION NAME DESCRIPTOR ADDRESS
00000018             0000    209          IDENT           = 24            ;SECTION IDENTIFIER
0000001C             0000    210          RELPAG          = 28            ;RELATIVE PAGE TO START MAPPING AT
00000020             0000    211          CHAN            = 32            ;CHANNEL THAT FILE TO MAP IS OPEN ON
00000024             0000    212          SECPAGCNT       = 36            ;NUMBER OF PAGES IN SECTION
00000028             0000    213          VBN             = 40            ;FILE VBN AT WHICH SECTION BEGINS
00000028             0000    214          PFN             = 40            ;STARTING PFN TO BE MAPPED
0000002C             0000    215          PROT            = 44            ;SECTION PROTECTION CODE
```

```
00000030  0000   216          PFC              = 48              ;SECTION PAGE FAULT CLUSTER
          0000   217 ;
          0000   218 ; OWN STORAGE:
          0000   219 ;
```

```
                                    I 1
0000    221          .SBTTL  CRMPSC - CREATE AND MAP SECTION
0000    222 ;++
0000    223 ; FUNCTIONAL DESCRIPTION:
0000    224 ;
0000    225 ;       THE CREATE AND MAP SECTION SERVICE CREATES A NEW SECTION
0000    226 ; OF THE SPECIFIED NAME UNLESS THAT NAME ALREADY EXISTS.  IN EITHER CASE
0000    227 ; IT THEN MAPS THE SPECIFIED SECTION.
0000    228 ;
0000    229 ; CALLING SEQUENCE:
0000    230 ;
0000    231 ;       CALLG   ARGLIST,@#SYS$CRMPSC
0000    232 ;
0000    233 ; INPUT PARAMETERS:
0000    234 ;
0000    235 ;       INADR(AP) = ADDRESS OF 2 LONG WORDS THE 1ST OF WHICH SPECIFIES
0000    236 ;               THE STARTING VIRTUAL ADDRESS TO CREATE, THE 2ND SPECIFIES
0000    237 ;               THE ENDING VIRTUAL ADDRESS TO CREATE (INCLUSIVE).
0000    238 ;       RETADR(AP) = ADDRESS OF A 2 LONGWORD ARRAY INTO WHICH IS RETURNED
0000    239 ;               THE STARTING AND ENDING VIRTUAL ADDRESSES (INCLUSIVE)
0000    240 ;               OF THE PAGES JUST CREATED
0000    241 ;       ACMODE(AP) = THE ACCESS MODE (MAXIMIZED WITH CALLING MODE)
0000    242 ;               USED AS THE OWNER OF THE NEW PAGE(S)
0000    243 ;       FLAGS(AP)  = BIT  0  - GBL - GLOBAL IF SET, PROCESS IF CLEAR
0000    244 ;                    BIT  1  - CRF - COPY ON REFERENCE
0000    245 ;                    BIT  2  - DZRO - DEMAND ZERO
0000    246 ;                    BIT  3  - WRT - WRITABLE IF SET, READ ONLY IF CLEAR
0000    247 ;                    BITS 4 - 5 RESERVED, MUST BE ZERO
0000    248 ;                    BITS 6 & 7 - WRTMOD - WRITE ACCESS FOR SECTION
0000    249 ;                    BITS 8 - 12 RESERVED, MUST BE ZERO
0000    250 ;                    BIT  13 - RESIDENT - RESIDENT GLOBAL SECTION
0000    251 ;                         BIT 13 IS NOT TO BE DOCUMENTED (WMC)
0000    252 ;                    BIT  14 - PERM - PERMANENT IF SET, TEMPORARY IF CLEAR
0000    253 ;                    BIT  15 - SYSGBL - SYSTEM GLOBAL IF SET, GROUP GLOBAL IF CLEAR
0000    254 ;                    BIT  16 - PFNMAP - MAP TO SPECIFIC PFN'S, IF SET
0000    255 ;                    BIT  17 - EXPREG - MAP TO FIRST FREE SPACE AVAILABLE
0000    256 ;                    BIT  18 - PROTECT - SET IF WRITE ACCESS MODE SPECIFIED
0000    257 ;                    BIT  19 - PAGFIL - USE PAGE FILE BACKING STORE FOR GLOBAL SECTI
0000    258 ;                    BIT  20 - EXECUTE - MAP IF EXECUTE ACCESS - EXEC MODE ONLY
0000    259 ;                    BITS 21 - 31 RESERVED, MUST BE ZERO
0000    260 ;       GSDNAM(AP) = THE DESCRIPTOR OF THE GLOBAL SECTION NAME
0000    261 ;       IDENT(AP)  = ADDRESS OF QUAD WORD CONTAINING SECTION IDENTIFICATION
0000    262 ;                    FIRST LONG WORD CONTAINS THE MATCH CONTROL INFORMATION
0000    263 ;                       0 = ISD$K_MATALL, MATCH ALWAYS
0000    264 ;                       1 = ISD$K_MATEQU, MATCH IF IDENTS ARE EQUAL
0000    265 ;                       2 = ISD$K_MATLEQ, MATCH IF HIGH 8 BITS ARE EQUAL
0000    266 ;                           AND LOW 24 BITS ARE LESS THAN OR EQUAL TO
0000    267 ;                           THE ID STORED IN THE GLOBAL SECTION.
0000    268 ;                    SECOND LONG WORD CONTAINS THE IDENT TO BE COMPARED
0000    269 ;       RELPAG(AP) = RELATIVE PAGE IN SECTION TO START MAPPING
0000    270 ;       CHAN(AP)   = CHANNEL ON WHICH FILE IS ACCESSED
0000    271 ;       SECPAGCNT(AP) = NUMBER OF PAGES (STARTING AT THE ABOVE VBN) IN THE SECTION
0000    272 ;       VBN(AP)    = STARTING VIRTUAL BLOCK IN FILE THAT BECOMES THE
0000    273 ;                    FIRST BLOCK OF THE GLOBAL SECTION
0000    274 ;       PFN(AP) = STARTING PFN TO BE MAPPED, IF PFNMAP FLAG SET
0000    275 ;       PROT(AP) = PROTECTION APPLIED TO SECTION
0000    276 ;       PFC(AP)    = PAGE FAULT CLUSTER
0000    277 ;
```

```
0000  ; IMPLICIT INPUTS:
0000  ;
0000  ;       NONE
0000  ;
0000  ; OUTPUT PARAMETERS:
0000  ;
0000  ;       NONE
0000  ;
0000  ; IMPLICIT OUTPUTS:
0000  ;
0000  ;       NONE
0000  ;
0000  ; COMPLETION CODES:
0000  ;
0000  ;       SS$_IVSECFLG - INVALID SECTION FLAGS
0000  ;       SS$_ACCVIO - ACCESS VIOLATION
0000  ;       SS$_CREATED - SECTION SUCCESSFULLY CREATED
0000  ;       SS$_NORMAL - EXISTING SECTION SUCESSFULLY MAPPED
0000  ;       SS$_ILLPAGCNT - PAGE COUNT LESS THAN ZERO
0000  ;       SS$_GSDFULL - NO FREE GSD TO ALLOCATE
0000  ;       SS$_IVSECIDCTL - INVALID SECTION IDENT MATCH CONTROL
0000  ;       SS$_GPTFULL - NO GLOBAL PAGE TABLE SPACE TO ALLOCATE
0000  ;       SS$_NOTFILEDEV - NOT A FILE ORIENTED DEVICE
0000  ;       SS$_IVCHNLSEC - INVALID CHANNEL FOR SECTION
0000  ;       SS$_ENDOFFILE - END OF FILE ENCOUNTERED MAPPING SECTION
0000  ;       SS$_VASFULL - VIRTUAL ADDRESS SPACE FULL
0000  ;       SS$_INSFMEM - INSUFFICIENT SHARED MEMORY GLOBAL PAGES AVAILABLE
0000  ;       SS$_INTERLOCK - UNABLE TO ACQUIRE SHARED MEMORY BIT MAP LOCK
0000  ;       SS$_TOOMANYLNAM - TOO MANY LOGICAL NAMES (DEPTH > 10)
0000  ;       SS$_IVLOGNAM - INVALID LOGICAL NAME
0000  ;       SS$_SHMNOTCNCT - SHARED MEMORY DATA STRUCTURES NOT CONNECTED
0000  ;       SS$_NOPRIV - NO PRIVILEGE FOR ATTEMPTED OPERATION
0000  ;       SS$_SECTBLFUL - SECTION TABLE FULL
0000  ;       SS$_PAGOWNVIO - PAGE OWNER VIOLATION (RE-MAPPING A PAGE IN USE)
0000  ;       SS$_IVCHAN - INVALID CHANNEL
0000  ;       SS$_EXQUOTA - EXCEEDED PAGING FILE QUOTA (FOR CRF SECTIONS)
0000  ;       SS$_NOWRT - UNABLE TO CREATE WRITABLE SECTION TO READ-ONLY FILE
0000  ;       SS$_IDMISMATCH - IDENT MISMATCH WITH EXISTING GLOBAL SECTION
0000  ;       SS$_EXPORTQUOTA - EXCEEDED PORT'S QUOTA FOR GSD'S
0000  ;       SS$_IVSSRQ - REQUEST TRIED TO MAP A MULTIPLE-PIECE SHARED
0000  ;               MEMORY GLOBAL SECTION BACKWARDS (THIS IS UN-IMPLEMENTED.)
0000  ;
0000  ; SIDE EFFECTS:
0000  ;
0000  ;       NONE
0000  ;
0000  ;--
0000  ;
0000  ; ***********************************************************************
0000  ;
0000  ; **************** THE FOLLOWING CODE MAY BE PAGED *****************
0000  ;
00000000        .PSECT  YF$$SYSCRMPSC
0000  ;
0000  ; ***********************************************************************
0000  ;
0000            .ENABL  LSB
```

```
                        0000
                        0000  ;
                        0000  ; NO INPUT RANGE ADDRESS SPECIFIED
                        0000  ;
                        0000  CRMPSC_NOINADR:
                        0000          ASSUME  SEC$V_GBL EQ 0
          0F 59    E9   0000          BLBC    R9,CRMPSC_ACCVIO          ;ONLY LEGAL FOR GLOBAL SECTIONS
    0B 59    0E    E1   0003          BBC     #SEC$V_PERM,R9,CRMPSC_ACCVIO ;WHICH ARE PERMANENT
             54    D7   0007          DECL    R4                        ;-1 INDICATES NO RANGE TO MAP
             21    11   0009          BRB     15$                       ;REJOIN CRMPSC CODE
                        000B
                        000B  FLAG_ERROR1:
 50    016C 8F    3C   000B          MOVZWL  #SS$_IVSECFLG,R0          ;SET INVALID SECTION FLAG ERROR CODE
             03    11   0010          BRB     CRMPSC_RET                ;BRANCH TO ERROR RETURN
                        0012  CRMPSC_ACCVIO:
       50    0C    3C   0012          MOVZWL  #SS$_ACCVIO,R0            ;CANNOT ACCESS THE RANGE TO MAP
                        0015  CRMPSC_RET:
                  04   0015          RET
                        0016
                  00000000          .PSECT  Y$EXEPAGED                ;PUT ENTRY POINT INTO SEPARATE PSECT
                        0000
                        0000  ;**********  EXE$CRMPSC  ENTRY POINT
                        0000
                 0FFC   0000          .ENTRY  EXE$CRMPSC,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
  00000016'EF    17   0002          JMP     EXE_CRMPSC                ;TRANSFER TO REAL PROCEDURE
                        0008
                  00000016          .PSECT  YF$$SYSCRMPSC             ;BACK TO $CRMPSC PSECT
                        0016
                        0016  EXE_CRMPSC:
       5E    1C    C2   0016          SUBL    S^#-MMG$C_LENGTH,SP       ;RESERVE A SCRATCH AREA
    59    10 AC    D0   0019          MOVL    FLAGS(AP),R9              ;GLOBAL SECTION FLAGS
    54    04 AC    D0   001D          MOVL    INADR(AP),R4              ;ADDRESS OF RANGE TO MAP
             DD    13   0021          BEQL    CRMPSC_NOINADR            ;BRANCH IF NULL RANGE
                        0023          IFNORD  #8,(R4),CRMPSC_ACCVIO     ;BRANCH IF RANGE NOT ACCESSIBLE
       54    64    7D   0029          MOVQ    (R4),R4                   ;PICK UP THE RANGE
             00    DD   002C  15$:    PUSHL   #0                        ;REMEMBER NO START RETADR YET
 7E    0619 8F    3C   002E          MOVZWL  #SS$_CREATED,-(SP)        ;ASSUME SECTION WILL BE CREATED
             30    BB   0033          PUSHR   #^M<R4,R5>                ;SAVE RANGE FOR MAPPING LATER
  00000000'GF    16   0035          JSB     G^MMG$RETADRINI           ;INIT RETURN RANGE, AND SCRATCH AREA
          D7 50    E9   003B          BLBC    R0,CRMPSC_RET             ;RETURN THE ERROR
       50    59    D0   003E          MOVL    R9,R0                     ;GLOBAL SECTION FLAGS
          FFBC'    30   0041          BSBW    MMG$VFYSECFLG             ;VERIFY SECTION FLAGS
       59    50    D0   0044          MOVL    R0,R9                     ;USE VERIFIED FLAGS
             58    D4   0047          CLRL    R8                        ;ASSUME NO SECTION TABLE ADDRESS
    56    20 AC    3C   0049          MOVZWL  CHAN(AP),R6               ;CHANNEL PARAMETER
             14    12   004D          BNEQ    17$                       ;BR ON MAPPED TO A FILE
    1E 59    13    E0   004F          BBS     #SEC$V_PAGFIL,R9,20$      ;SKIP CHANNEL CHECKS IF PAGE FILE
    B4 59    10    E1   0053          BBC     #SEC$V_PFNMAP,R9,FLAG_ERROR1 ;ERR IF NOT PFNMAP AND NO CHANNEL
             18    11   0057          BRB     20$                       ;SKIP CHANNEL VERIFICATION AS NO CHANNEL
                        0059
                        0059  CRMPSC_FOUND:                             ;GLOBAL SECTION EXISTS, JUST MAP TO IT
       5E    34    C0   0059          ADDL    #<11*4>+8,SP              ;CLEAN OFF THE STACK
    08 AE    01    3C   005C          MOVZWL  #SS$_NORMAL,8(SP)         ;INDICATE SECTION ALREADY EXISTS
          041A    31   0060          BRW     MAPGBLSEC1                ;GO MAP THE SECTION
                        0063
    A4 59    10    E0   0063  17$:    BBS     #SEC$V_PFNMAP,R9,FLAG_ERROR1    ;BR IF PFNMAP, ERROR
    A0 59    13    E0   0067          BBS     #SEC$V_PAGFIL,R9,FLAG_ERROR1    ;BR IF PAGFIL BACKING STORE, ERR
```

```
                                006B
                                006B  ; CALL TO IOC$VERIFYCHAN THAT WAS MADE HERE IS NOW BEING MADE IN CHECK_WINDOW
                                006B
                  0D2A    30    006B  19$:    BSBW    CHECK_WINDOW            ;INSURE THAT FILE IS COMPLETELY MAPPED
                 A4 50    E9    006E          BLBC    R0,CRMPSC_RET           ;BRANCH IF FILE NOT FULLY MAPPED
     54    00000000'GF    D0    0071  20$:    MOVL    G^SCH$GL_CURPCB,R4      ;GET PCB ADDRESS
            05 59    01    E1    0078          BBC     #SEC$V_CRF,R9,25$       ;BRANCH IF NOT CRF SECTION
        00 FC AD     08    E2    007C          BBSS    S^#MMG$V_CHGPAGFIL,B^MMG$L_MAXACMODE(FP),25$ ;MUST CHARGE PAGE FILE
                                0081
                                0081          ASSUME  SEC$V_GBL EQ 0
                 55 59    E9    0081  25$:    BLBC    R9,CRMPSC_PROCESS       ;BRANCH IF CREATING PROCESS SECTION
     50    00000000'GF    DE    0084          MOVAL   G^EXE$GL_GSDMTX,R0      ;ADR OF GLOBAL SECTION MUTEX
           00000000'GF    16    008B          JSB     G^SCH$LOCKW             ;LOCK GSD MUTEX FOR WRITING
                                0091  ;
                                0091  ; RETURNS WITH IPL = ASTDEL
                                0091  ; R8 = CHANNEL CONTROL BLOCK ADDRESS, IF PFNMAP CLEAR
                                0091  ; R9 = SECTION FLAGS
                                0091  ; 0(SP)  = STARTVA
                                0091  ; 4(SP) = ENDVA
                                0091  ; 8(SP) = SUCCESS CODE FOR MAP GLOBAL SECTION
                                0091  ;
                                0091  ; ***** NOTE:  ALL EXITS FROM THIS POINT ON MUST UNLOCK THE GSD MUTEX.
                                0091  ;
                                0091  ; FIRST, BEFORE DOING A GSD SCAN, DO A DEALLOCATE SECTION SCAN.  THIS
                                0091  ; SCAN MUST BE DONE IN ORDER TO COMPLETE THE DELETION OF A TEMPORARY
                                0091  ; GLOBAL SECTION WHOSE REFERENCE COUNT HAS DROPPED TO ZERO.
                                0091  ;
           00000000'EF    16    0091          JSB     MMG$DALCSTXSCN1         ;SCAN FOR SECTIONS TO DEALLOCATE --
                                0097  ;
                 5E    34    C2    0097          SUBL    #<11*4>+8,SP            ;MAKE ROOM FOR GSD NAME AND IDENT
                 57    5E    D0    009A          MOVL    SP,R7                   ;R7 = ADDRESS OF GSD NAME STORAGE
                                009D
                                009D          ASSUME  IDENT EQ GSDNAM+4
            50    14 AC    7D    009D          MOVQ    GSDNAM(AP),R0           ;DESCRIPTOR ADDRESS FOR GLOBAL SECTION NAME
                                00A1                                          ;ADDRESS OF IDENTIFICATION QUAD WORD
                 56    59    D0    00A1          MOVL    R9,R6                   ;SECTION FLAGS
                    FF59'    30    00A4          BSBW    MMG$GSDSCN              ;SCAN FOR THE NAME
                                00A7  ;
                                00A7  ; R4 = SHARED MEMORY CONTROL BLOCK ADR IF GLOBAL SECTION IS IN SHARED MEMORY
                                00A7  ; R5 RETURNED WITH SYSTEM PROCESS HEADER ADDRESS WHETHER SUCCESSFUL OR NOT
                                00A7  ; R10 =   0 IF THE GSD WAS FOUND IN LOCAL MEMORY
                                00A7  ;         -1 IF THE LOCAL MEMORY SEARCH EXTENDED INTO SHARED MEMORY TABLES
                                00A7  ;         >0 IF A SPECIFIC SHARED MEMORY NAME WAS SPECIFIED
                                00A7  ;
                 AF 50    E8    00A7          BLBS    R0,CRMPSC_FOUND         ;BRANCH IF SECTION ALREADY EXISTS
        50    0978 8F    B1    00AA          CMPW    #SS$_NOSUCHSEC,R0       ;CONTINUE WITH CREATION IF NOT FOUND
                    35    12    00AF          BNEQ    ULKGSDMTXRET1           ;BRANCH IF SOME OTHER ERROR
                                00B1  ;
                                00B1  ; GLOBAL SECTION NAME DOESN'T EXIST, CREATE A NEW ONE
                                00B1  ; ALLOCATE A GLOBAL SECTION DESCRIPTOR BLOCK
                                00B1  ;  0(SP) - 43(SP) = GSD NAME COUNTED STRING
                                00B1  ; 44(SP) - 51(SP) = IDENTIFICATION QUAD WORD
                                00B1  ;
                                00B1
                                00B1          ASSUME  PRV$V_SYSGBL EQ PRV$V_PRMGBL+1
                                00B1          ASSUME  SEC$V_SYSGBL EQ SEC$V_PERM+1
     5B    00000000'9F    D0    00B1          MOVL    @#CTL$GL_PHD,R11        ;YES, THEN CHECK FOR PRIVILEGE TO DO SO
```

```
                50  6B  02    18  EF  00B8   449          EXTZV   #PRV$V_PRMGBL,#2,(R11),R0 ;GET PRIVILEGES FOR PERM AND SYS
                51  59  02    0E  EF  00BD   450          EXTZV   #SEC$V_PERM,#2,R9,R1    ;GET DESIRED PERMANENT AND SYSTEM WIDE FLAGS
                        51    50  CA  00C2   451          BICL    R0,R1                   ;IF ANY DESIRED BITS STILL LEFT ON
                              1C  12  00C5   452          BNEQ    CRMPSC_NOPRIV           ;THEN NO PRIV FOR REQUESTED OPERATION
                    04  59    10  E1  00C7   453          BBC     #SEC$V_PFNMAP,R9,26$    ;IS PFNMAPPING REQUESTED?
                    14  6B    1A  E1  00CB   454          BBC     #PRV$V_PFNMAP,PHD$Q_PRIVMSK(R11),CRMPSC_NOPRIV ;NEEDS PRIV
                    23  59    0D  E1  00CF   455 26$:      BBC     #SEC$V_RESIDENT,R9,30$  ;IS RESIDENT SECTION REQUESTED?
                    1F  6B    00  E0  00D3   456          BBS     #PRV$V_CMKRNL,PHD$Q_PRIVMSK(R11),30$ ;NEEDS PRIV
                              0A  11  00D7   457          BRB     CRMPSC_NOPRIV           ;REPORT NO PRIV TO DO PFNMAP
                                      00D9   458 CRMPSC_PROCESS:                      ;BRANCH ASSIST TO PROCESS SECTION MAP
                          0A8F    31  00D9   459          BRW     MAP_PROCESS_SEC         ;GO MAP PROCESS SECTION
                                      00DC   460 ;
                                      00DC   461 ; ERROR IN SECTION FLAGS AND TYPE OF SECTION REQUESTED
                                      00DC   462 ;
                                      00DC   463 FLAG_ERROR:
                50      016C  8F    3C  00DC   464          MOVZWL  #SS$_IVSECFLG,R0        ;REPORT INVALID SECTION FLAGS
                              03    11  00E1   465 28$:     BRB     ULKGSDMTXRET1           ;RETURN ERROR STATUS TO CALLER
                                      00E3   466 ;
                                      00E3   467 ; NO PRIVILEGE TO CREATE OR MAP THE GLOBAL SECTION
                                      00E3   468 ;
                                      00E3   469 CRMPSC_NOPRIV:
                        50    24    3C  00E3   470          MOVZWL  #SS$_NOPRIV,R0          ;NO PRIVILEGE FOR REQUESTED OPERATION
                                      00E6   471 ULKGSDMTXRET1:
                          0328    31  00E6   472          BRW     ULKGSDMTXRET
                                      00E9   473 ILL_PAG_CNT:
                    50      FC  8F    9A  00E9   474          MOVZBL  #SS$_ILLPAGCNT,R0       ;REPORT SECTION HAS NO PAGES IN IT
                              F7    11  00ED   475          BRB     ULKGSDMTXRET1           ;RETURN TO CALLER
                                      00EF   476 SHM_NOT_MAP:
            50      036C  8F    3C  00EF   477          MOVZWL  #SS$_SHMGSNOTMAP,R0     ;REPORT SECTION IS NOT BEING MAPPED
                              F0    11  00F4   478          BRB     ULKGSDMTXRET1           ;RETURN TO CALLER
                                      00F6   479                                          ;R5=SYSTEM PROCESS HEADER ADR
                57      24  AC    D0  00F6   480 30$:     MOVL    SECPAGCNT(AP),R7        ;SET COUNT OF PAGES IN SECTION
                            ED    19  00FA   481          BLSS    ILL_PAG_CNT             ;ERROR, ILLEGAL PAGE COUNT
                            5A    D5  00FC   482          TSTL    R10                     ;IS SECTION IN SHARED MEMORY?
                            52    15  00FE   483          BLEQ    LOCAL_MEM_GSD           ;BR ON NO
                            57    D5  0100   484          TSTL    R7                      ;IS A NON-ZERO PAGECOUNT SPECIFIED?
                            E5    13  0102   485          BEQL    ILL_PAG_CNT             ;BR ON NO, THIS IS NOT IMPLEMENTED
                DB  6B      1B    E1  0104   486          BBC     #PRV$V_SHMEM,PHD$Q_PRIVMSK(R11),CRMPSC_NOPRIV ;SHMEM PRIV NEEDED
                D0  59      01    E0  0108   487          BBS     #SEC$V_CRF,R9,FLAG_ERROR  ;SHARED MEMORY CAN'T BE CRF
                CC  59      10    E0  010C   488          BBS     #SEC$V_PFNMAP,R9,FLAG_ERROR     ;SHMEM CANNOT BE PFNMAP
                C8  59      13    E0  0110   489          BBS     #SEC$V_PAGFIL,R9,FLAG_ERROR     ;SHMEM CANNOT ALSO BE PAGFIL
                C4  59      0E    E1  0114   490          BBC     #SEC$V_PERM,R9,FLAG_ERROR ;SHARED MEMORY MUST BE PERMANENT
                00  59      04    E2  0118   491          BBSS    #SEC$V_SHMGS,R9,31$     ;INDICATE SECTION IS IN SHARED MEMORY
                10  59      02    E1  011C   492 31$:     BBC     #SEC$V_DZRO,R9,32$      ;IF NOT DZRO, THEN NEED NOT BE MAPPED
                        1C  AC    D5  0120   493          TSTL    RELPAG(AP)              ;FOR SH MEM GS, ALL SECTION MUST BE
                            CA    12  0123   494          BNEQ    SHM_NOT_MAP             ;MAPPED OTHERWISE IT CAN'T BE INITED
                50      38  AE    D0  0125   495          MOVL    <<1T*4>-8+4>(SP),R0     ;THIS MEANS RELPAG MUST BE ZERO AND
                            C4    13  0129   496          BEQL    SHM_NOT_MAP             ;THE INPUT ADDRESS RANGE MUST SPECIFY
                    50      50    D2  012B   497          MCOML   R0,R0                   ;A VALID RANGE OF ADDRESSES (MINUS
                            BF    13  012E   498          BEQL    SHM_NOT_MAP             ;ONE AND ZERO ARE NOT VALID)
                                      0130   499                                          ;R4=SHARED MEMORY CONTROL BLOCK ADR
                      FECD'     30  0130   500 32$:     BSBW    MMG$ALOSHMGSD           ;ALLOCATE A SHARED MEMORY GSD
                        AB  50    E9  0133   501          BLBC    R0,28$                  ;BR IF UNABLE TO GET A GSD
                                      0136   502                                          ;R4=SHARED MEMORY CONTROL BLOCK ADR
                      FEC7'     30  0136   503          BSBW    MMG$ALOSHMPAG           ;ALLOCATE PAGES OF SHARED MEMORY
                        6E  50    E9  0139   504          BLBC    R0,RELEAS_GSD_RET       ;BR IF COULDN'T GET ENOUGH PAGES
                50      22  A6    9E  013C   505          MOVAB   GSD$T_GSDNAM(R6),R0     ;GET ADDRESS OF GSD ASCIC NAME FIELD
```

```
                                                     N 1
                51    5E   D0  0140   506          MOVL    SP,R1                   ;POINT TO ASCIC NAME STRING TO STORE
                      2D   10  0143   507          BSBB    MOVGSDNAM               ;MOVE GLOBAL SECTION NAME INTO GSD
                      58   D5  0145   508          TSTL    R8                      ;IS GS MAPPED TO A FILE?
                      03   12  0147   509          BNEQ    COMMON_INIT1            ;BR IF MAPPED TO A FILE
                   00D6    31  0149   510          BRW     NO_FILE_INIT            ;BR IF NOT MAPPED TO A FILE
                           014C   511 ;
                           014C   512 COMMON_INIT1:
                   00E0    31  014C   513          BRW     COMMON_INIT             ;SKIP EXTGSD SPECIFIC INITIALIZATION
                           014F   514 PFNMAP_GSD_1:
                   0096    31  014F   515          BRW     PFNMAP_GSD
                           0152   516 ;
                           0152   517 LOCAL_MEM_GSD:
             F9 59    10    E0  0152   518          BBS     #SEC$V_PFNMAP,R9,PFNMAP_GSD_1 ;BR IF PFNMAP AS GSD IS LARGER
                51    6E   9A  0156   519          MOVZBL  (SP),R1                 ;GET SIZE OF GLOBAL SECTION NAME
                51    23   C0  0159   520          ADDL2   #GSD$C_LENGTH,R1        ;CALCULATE TOTAL GSD STRUCTURE SIZE
                      51   DD  015C   521          PUSHL   R1                      ;SAVE GSD SIZE FOR LATER
          00000000'GF    16  015E   522          JSB     G^EXE$ALOPAGED          ;ALLOCATE FROM PAGED POOL
                51    8E   D0  0164   523          MOVL    (SP)+,R1                ;RESTORE GSD SIZE FOR LATER
                   60 50    E8  0167   524          BLBS    R0,NORMAL_GSD           ;BRANCH IF ONE AVAILABLE
           50    00CC 8F    3C  016A   525          MOVZWL  #SS$_GSDFULL,R0         ;GLOBAL SECTION DESCRIPTOR TABLE FULL
                   029F    31  016F   526 37$:      BRW     ULKGSDMTXRET            ;UNLOCK THE MUTEX AND RETURN ERROR
                           0172   527 ;
                           0172   528 MOVGSDNAM:
                      3C   BB  0172   529          PUSHR   #^M<R2,R3,R4,R5>        ;SAVE SOME REGS TO BE SAFE
                52    51   D0  0174   530          MOVL    R1,R2                   ;NAME POINTER FOR HASH
                53    82   9A  0177   531          MOVZBL  (R2)+,R3                ;NAME LENGTH
                      54   D4  017A   532          CLRL    R4                      ;HASH ACCUMULATOR
                54    82   80  017C   533 10$:     ADDB    (R2)+,R4                ;CALCULATE A SIMPLE HASH
                   FA 53    F5  017F   534          SOBGTR  R3,10$
          0B A6    54    90  0182   535          MOVB    R4,GSD$B_HASH(R6)       ;SAVE IT
                52    81   9B  0186   536          MOVZBW  (R1)+,R2                ;GET GLOBAL SECTION NAME LENGTH
                80    52   90  0189   537          MOVB    R2,(R0)+                ;STORE NAME LENGTH IN GSD
           60   61    52   28  018C   538          MOVC3   R2,(R1),(R0)            ;MOVE NAME STRING INTO GSD
                      3C   BA  0190   539          POPR    #^M<R2,R3,R4,R5>        ;RESTORE THE REGISTERS
                           05  0192   540          RSB
                           0193   541 ;
                           0193   542 RELEAS_PAG_RET:
                50    01   9A  0193   543          MOVZBL  #SHD$V_BITMAPLCK,R0     ;NUMBER OF BIT LOCK REQUESTED
                   FE67'    30  0196   544          BSBW    MMG$SHMTXLK             ;REQUEST MUTEX AND BIT LOCK
                           0199   545 ; **********
                           0199   546 ; AT SOME TIME THIS SHOULD SEND A MESSAGE TO THE ERROR LOGGER.
                           0199   547 ; **********
                   10 50    E9  0199   548          BLBC    R0,34$                  ;UNABLE TO GET BIT LOCK
                   FE61'    30  019C   549          BSBW    MMG$SET_BITMAP          ;RELEASE ALL PAGES ALLOCATED FOR GS
          00 009F C5    01   E7  019F   550          BBCCI   #SHD$V_BITMAPLCK,SHD$B_FLAGS(R5),33$ ;RELEASE BIT LCK
                   FE58'    30  01A5   551 33$:     BSBW    MMG$SHMTXULK            ;RELEASE SHM MUTEX
                      02    11  01A8   552          BRB     34$                     ;JOIN COMMON CODE
                           01AA   553 RELEAS_GSD_RET:                              ;R0 CONTAINS ERROR CODE
                      50   DD  01AA   554          PUSHL   R0                      ;REMEMBER RETURN STATUS CODE
                50    01   9A  01AC   555 34$:     MOVZBL  #1,R0                   ;ONE REFERENCE COUNT THAT LOCKS GSD
           00000000'GF    16  01AF   556          JSB     G^MMG$DECSHMREF         ;IS GIVEN BACK
            00 66    01   E7  01B5   557          BBCCI   #GSD$V_LOCKED,GSD$L_GSDFL(R6),35$ ;UNLOCK THE GSD
            51   15 A4    9A  01B9   558 35$:     MOVZBL  SHB$B_PORT(R4),R1       ;GET PORT # FOR THIS PROCESSOR
                 0C A4    D7  01BD   559          DECL    SHB$L_REFCNT(R4)        ;ONE LESS GSD OWNED BY THIS PORT
          3C A541    01    58  01C0   560          ADAWI   #1,SHD$W_GSDQUOTA(R5)[R1] ;GIVE SHM GSD QUOTA BACK TO PORT
                50    8E   D0  01C5   561          MOVL    (SP)+,R0                ;RESTORE RETURN STATUS CODE
                      A5   11  01C8   562          BRB     37$                     ;ERROR RETURN
```

```
                                                     B 2
                     01CA   563 NORMAL_GSD:
      56    52   D0  01CA   564          MOVL    R2,R6                           ;ADDRESS OF NEW GSD
   08 A6    51   B0  01CD   565          MOVW    R1,GSD$W_SIZE(R6)               ;FILL IN SIZE OF GSD
   0A A6    15   90  01D1   566          MOVB    #DYN$C_GSD,GSD$B_TYPE(R6) ;AND TYPE CODE
   50    22 A6   9E  01D5   567          MOVAB   GSD$T_GSDNAM(R6),R0             ;LOAD POINTER TO ASCIC NAME FIELD IN GSD
      51    5E   D0  01D9   568          MOVL    SP,R1                           ;COPY POINTER TO ASCIC NAME STRING
            94   10  01DC   569          BSBB    MOVGSDNAM                       ;MOVE GLOBAL SECTION NAME INTO GSD
            4F   11  01DE   570          BRB     COMMON_INIT                     ;SKIP EXTGSD SPECIFIC INITIALIZATION
                     01E0   571 ;
50    00CC 8F   3C  01E0   572 36$:     MOVZWL  #SS$_GSDFULL,R0                 ;GLOBAL SECTION DESCRIPTOR TABLE FULL
         0229   31  01E5   573          BRW     ULKGSDMTXRET                    ;UNLOCK THE MUTEX AND RETURN ERROR
                     01E8   574 ;
                     01E8   575 PFNMAP_GSD:
      51    6E   9A  01E8   576          MOVZBL  (SP),R1                         ;GET SIZE OF GLOBAL SECTION NAME STRING
      51    31   C0  01EB   577          ADDL2   #GSD$C_EXTGSDLNG,R1             ;CALCULATE TOTAL SIZE OF EXTENDED GSD
            51   DD  01EE   578          PUSHL   R1                              ;SAVE GSD SIZE FOR LATER
00000000'GF      16  01F0   579          JSB     G^EXE$ALOPAGED                  ;ALLOCATE FROM PAGED POOL
      51    8E   D0  01F6   580          MOVL    (SP)+,R1                        ;RESTORE GSD SIZE
        E4  50   E9  01F9   581          BLBC    R0,36$                          ;BRANCH IF NONE AVAILABLE
      56    52   D0  01FC   582          MOVL    R2,R6                           ;REMEMBER ADDRESS OF NEW EXTGSD
   08 A6    51   B0  01FF   583          MOVW    R1,GSD$W_SIZE(R6)               ;FILL IN SIZE OF EXTGSD
   0A A6    28   90  0203   584          MOVB    #DYN$C_EXTGSD,GSD$B_TYPE(R6) ;AND TYPE CODE
   50    30 A6   9E  0207   585          MOVAB   GSD$T_PFNGSDNAM(R6),R0          ;POINT TO DESTINATION OF NAME STRING
      51    5E   D0  020B   586          MOVL    SP,R1                           ;POINT TO ASCIC NAME STRING ITSELF
         FF61   30  020E   587          BSBW    MOVGSDNAM                       ;MOVE GLOBAL SECTION NAME INTO GSD
      2C A6    D4  0211   588          CLRL    GSD$L_REFCNT(R6)                ;INITIALIZE REFERENCE TO SECTION
24 A6    28 AC   D0  0214   589          MOVL    PFN(AP),GSD$L_BASEPFN(R6) ;FILL IN STARTING PFN FOR SECTION
   28 A6    57   D0  0219   590          MOVL    R7,GSD$L_PAGES(R6)              ;AND NUMBER OF PAGES IN SECTION
            03   12  021D   591          BNEQ    NO_FILE_INIT                    ;BR IF PAGE COUNT VALID
         FEC7   31  021F   592          BRW     ILL_PAG_CNT                     ;BR IF BAD PAGE COUNT
                     0222   593 NO_FILE_INIT:
14 A6    2C AC   B0  0222   594          MOVW    PROT(AP),GSD$W_PROT(R6) ;AND PROTECTION FOR SECTION
         16 A6   B4  0227   595          CLRW    GSD$W_GSTX(R6)                  ;SET NO SECTION TABLE INDEX
            5A   D4  022A   596          CLRL    R10                             ;REMEMBER NO SECTION TABLE ENTRY
         10 A6   D4  022C   597          CLRL    GSD$L_FILUIC(R6)                ;0 INDICATES BYPASS FILE PROT CHECK
                     022F   598 COMMON_INIT:
50    00000000'GF   D0  022F   599          MOVL    G^SCH$GL_CURPCB,R0              ;GET PCB ADDRESS
 0C A6    00BC C0   D0  0236   600          MOVL    PCB$L_UIC(R0),GSD$L_PCBUIC(R6) ;SET UIC OF CREATOR FROM PCB
      5E    2C   C0  023C   601          ADDL2   #<11*4>,SP                      ;CLEAN GS NAME OFF STACK
      50    8E   7D  023F   602          MOVQ    (SP)+,R0                        ;R0=MATCH CONTROL, R1=IDENTIFICATION DATA
   08    FC AD   F0  0242   603          INSV    B^MMG$L_MAXACMODE(FP),#SEC$V_AMOD,- ;SET THE ACCESS
      59    02         0246   604                  #SEC$S_AMOD,R9                  ;MODE OF THE SECTION
   20 A6    59   B0  0248   605          MOVW    R9,GSD$W_FLAGS(R6)              ;FILL IN SECTION FLAGS
      02    50   D1  024C   606          CMPL    R0,#SEC$K_MATLEQ                ;VALID MATCH CONTROL
            61   14  024F   607          BGTR    BADMATCHCTL                     ;BRANCH IF NOT
   18 A6    51   D0  0251   608          MOVL    R1,GSD$L_IDENT(R6)              ;SET GLOBAL SECTION IDENT
   50 59    10   E0  0255   609          BBS     #SEC$V_PFNMAP,R9,BRW_70$  ;BR IF NO SECTION TABLE ENTRY
                     0259   610 ;
                     0259   611 ; ALLOCATE A GLOBAL SECTION TABLE ENTRY FOR THE NEW SECTION
                     0259   612 ; R4 = SHARED MEMORY CONTROL BLOCK ADDRESS IF GLOBAL SECTION IS IN SHARED MEM
                     0259   613 ; R5 = SYSTEM PROCESS HEADER ADDRESS
                     0259   614 ; R6 = GLOBAL SECTION DESCRIPTOR ADDRESS
                     0259   615 ; R7 = COUNT OF PAGES IN SECTION
                     0259   616 ; R8 = CHANNEL CONTROL BLOCK ADDRESS, OR
                     0259   617 ;      = 0 IF PFNMAP OR PAGFIL SECTION
                     0259   618 ; R9 = SECTION FLAGS
                     0259   619 ; R10 = SECTION TABLE ENTRY ADDRESS, OR
```

```
                                0259   620 ;       = 0 IF PFNMAP SECTION
                                0259   621 ; 0(SP) = STARVA
                                0259   622 ; 4(SP) = ENDVA
                                0259   623 ; 8(SP) = SUCCESS CODE FOR MAP GLOBAL SECTION
                                0259   624 ;
                                0259   625 MAP_TO_FILE:
                     0801   30  0259   626         BSBW    INITSECTBL              ;ALLOCATE AND INIT SECTION TABLE
                                025C   627                                         ;R2 AND R3 DESTROYED
                  58 50     E9  025C   628         BLBC    R0,GSTFULL1             ;BRANCH IF NONE AVAILABLE
            20 A6 14 AA     B0  025F   629         MOVW    SEC$W_FLAGS(R10),GSD$W_FLAGS(R6) ;RECORD FLAGS IN GSD
                                0264   630 ;
                                0264   631 ; R1  = SECTION TABLE INDEX
                                0264   632 ; R7  = SECTION PAGE COUNT
                                0264   633 ; R10 = SECTION TABLE BASE ADDRESS
                                0264   634 ;
               16 A6    51  B0  0264   635         MOVW    R1,GSD$W_GSTX(R6)       ;SAVE SECTION INDEX
                  10 A6 D4      0268   636         CLRL    GSD$L_FILUIC(R6)        ;ASSUME NO FILE PROTECTION CHECK
               50 0C AA D0      026B   637         MOVL    SEC$L_WINDOW(R10),R0    ;GET WINDOW ADDRESS
                     51 13      026F   638         BEQL    392$                    ;NO WINDOW - PAGE FILE BACKING STORE
               50 18 A0 D0      0271   639         MOVL    WCB$L_FCB(R0),R0        ;CHAIN TO FCB ADDRESS
                     08 13      0275   640         BEQL    38$                     ;BRANCH IF NOT AN FCP WINDOW
            70 A0 2C AC A9      0277   641         BISW3   PROT(AP),FCB$W_FILEPROT(R0),- ;SET POSSIBLY RESTRICTED
                  14 A6         027C   642                 GSD$W_PROT(R6)          ;FILE PROTECTION
               50 58 A0 D0      027E   643         MOVL    FCB$L_FILEOWNER(R0),R0  ;AND FETCH THE FILE'S OWNER
               10 A6    50 D0   0282   644 38$:    MOVL    R0,GSD$L_FILUIC(R6)     ;SET FILE OWNER INTO GSD
                                0286   645                                         ;0 INDICATES BYPASS FILE PROT CHECK
               29 0A A6 91      0286   646         CMPB    GSD$B_TYPE(R6),#DYN$C_SHMGSD ;IS GS IN SHARED MEMORY?
                     20 13      028A   647 39$:    BEQL    SHM_GS_MAP              ;BR ON YES, GO READ SECTION INTO MEMORY
                                028C   648 ;
                                028C   649 ; GET GLOBAL PAGE TABLE ENTRIES
                                028C   650 ;
            51 57    03 C1      028C   651 391$:   ADDL3   #3,R7,R1                ;PAGCNT + 2 FOR STOPPER ENTRIES
                  51 01 CA      0290   652         BICL    #1,R1                   ;ROUND UP TO EVEN NO. OF LONG WORDS
                  51 04 C4      0293   653         MULL    #4,R1                   ;DESIRED NO. OF BYTES OF GPTE
         53 00000000'GF DE      0296   654         MOVAL   G^EXE$GL_GPT,R3         ;GPT ALLOCATION LIST HEAD
            00000000'GF 16      029D   655         JSB     G^EXE$ALLOCATE          ;GET SOME GPT
                  40 50 E8      02A3   656         BLBS    R0,40$                  ;BRANCH IF GOT SOME
                   0130 31      02A6   657         BRW     GPTFULL                 ;GLOBAL PAGE TABLE FULL
                                02A9   658
                                02A9   659 BRW_70$:
                   00A2 31      02A9   660         BRW     70$                     ;BRANCH ASSIST
                                02AC   661 SHM_GS_MAP:
                   18 AA D4     02AC   662         CLRL    SEC$L_REFCNT(R10)       ;DON'T USE SEC TBL REFCNT FOR SH MEM GS
                    0200 31     02AF   663         BRW     MAPGBLSEC2              ;GO MAP THE SECTION
                                02B2   664 ;
                                02B2   665 ;
                                02B2   666 ; BAD MATCH CONTROL FIELD FOR SECTION
                                02B2   667 ;
                                02B2   668 BADMATCHCTL:                         ;BAD MATCH CONTROL FIELD
            50 02E4 8F      3C  02B2   669         MOVZWL  #SS$_IVSECIDCTL,R0      ;INVALID SECTION IDENT CONTROL
                                02B7   670 GSTFULL1:
               28 0A A6 91      02B7   671         CMPB    GSD$B_TYPE(R6),#DYN$C_EXTGSD ;IS THIS AN EXTENDED GSD?
                     26 1B      02BB   672         BLEQU   BRGSTFULL               ;BR IF LOCAL MEMORY OR EXTENDED GSD
                     50 DD      02BD   673         PUSHL   R0                      ;SAVE ERROR STATUS CODE
                   FED1 31      02BF   674         BRW     RELEAS_PAG_RET          ;BR IF SHARED MEMORY GSD
                                02C2   675 ;
                                02C2   676 ; CHECK FOR PAGE FILE LIMIT EXCEEDED AND FILL IN PROTECTION
```

```
                                02C2    677 ;
            14 A6    2C AC   B0 02C2    678 392$:   MOVW    PROT(AP),GSD$W_PROT(R6) ;PROTECTION REQUESTED FOR THE GLOBAL SECTION
  00000000'EF    1C AA   C2 02C7    679         SUBL    SEC$L_PAGCNT(R10),MMG$GL_GBLPAGFIL
                    BB   18 02CF    680         BGEQ    391$                    ;NO PROBLEMS
     50 00000000'EF    1C AA   C1 02D1    681         ADDL3   SEC$L_PAGCNT(R10),MMG$GL_GBLPAGFIL,R0
          00002164 8F   DD 02DA    682         PUSHL   #SS$_EXGBLPAGFIL
                 00FC   31 02E0    683         BRW     FRESTXERR               ;THIS WILL ALSO FIX UP THE LIMIT
                                02E3    684 ;
                 011E   31 02E3    685 BRGSTFULL: BRW  GSTFULL
                                02E6    686 ;
                                02E6    687 ;
                                02E6    688 ; INITIALIZE GLOBAL PAGE TABLE ENTRIES
                                02E6    689 ;
                                02E6    690 ; R2 = ADDRESS OF FIRST GPTE ALLOCATED, FRONT STOPPER GPTE
                                02E6    691 ; R7 = DESIRED SECTION PAGE COUNT
                                02E6    692 ; R10 = SECTION TABLE ENTRY ADDRESS
                                02E6    693 ;
                    82   D4 02E6    694 40$:    CLRL    (R2)+                   ;SET FRONT STOPPER FOR GLOBAL SECTION
                 53    52   D0 02E8    695         MOVL    R2,R3                   ;FIRST GPTE ADDRESS
           52 00000000'EF   C2 02EB    696         SUBL    MMG$GL_GPTBASE,R2       ;BYTE OFFSET FROM BEGINNING OF GPT
                 52    04   C6 02F2    697         DIVL    #4,R2                   ;LONG WORD OFFSET = GPTX
              08 AA    52   C8 02F5    698         BISL    R2,SEC$L_VPXPFC(R10)    ;PUT GPTX IN (PFC ALREADY PRESENT)
                                02F9    699 ;
                                02F9    700 ; R3 = ADDRESS OF FIRST GLOBAL PTE TO FILL IN
                                02F9    701 ; R5 = PROCESS HEADER ADDRESS FOR SYSTEM
                                02F9    702 ; R7 = SECTION PAGE COUNT
                                02F9    703 ;
           52 01000000 8F   D0 02F9    704         MOVL    #2@PTE$V_OWN,R2         ;GLOBAL DZRO FORMAT FOR PAGE FILE BACKING ST
              18 59    13   E0 0300    705         BBS     #SEC$V_PAGFIL,R9,45$    ;PAGE FILE BACKING STORE
                                0304    706         ASSUME  SEC$V_DZRO EQ SEC$V_CRF+1 ;REQUIRE BITS TO BE ADJACENT
                                0304    707         ASSUME  SEC$V_WRT EQ SEC$V_DZRO+1
     52    14 AA    03    01   EF 0304    708         EXTZV   #SEC$V_CRF,#3,SEC$W_FLAGS(R10),R2 ;CRF, DZRO, AND WRT BITS
              52    0540 8F   A8 030A    709         BISW    #<PTE$M_TYP1 ! PTE$M_TYP0 !- ;SECTION TYPE PTE
                                030F    710                 <2@PTE$V_OWN>>@-16,R2   ;OWNER FIELD USED FOR GLOBAL BITS
                                030F    711         ASSUME  GSD$W_GSTX GE 2
                 51    14 A6   D0 030F    712         MOVL    GSD$W_GSTX-2(R6),R1     ;HIGH 16 BITS OF R1 = SEC TBL INDEX
              51    51    10   79 0313    713         ASHQ    #16,R1,R1               ;R2 = PAGE TABLE ENTRY
           04 14 AA    03   E1 0317    714         BBC     #SEC$V_WRT,SEC$W_FLAGS(R10),50$ ;BRANCH IF NOT WRITABLE
              00 52    17   E2 031C    715 45$:    BBSS    #PTE$V_OWN,R2,50$       ;SET GBLWRT BIT IN OWNER FIELD
                                0320    716 50$:
                                0320    717 ;
                                0320    718 ; R7 = COUNT OF GLOBAL PAGE TABLE ENTRIES TO LOCK AND FILL IN
                                0320    719 ; R2 = PAGE TABLE ENTRY TO STORE IN NEW GPTE'S
                                0320    720 ; R3 = ADDRESS OF FIRST GLOBAL PAGE TABLE ENTRY
                                0320    721 ;
                 0088 8F   BB 0320    722         PUSHR   #^M<R3,R7>
                 63    08   90 0324    723 60$:    MOVB    #IPL$_SYNCH,(R3)        ;SET UP TO REFERENCE PAGE AND
           03D5'CF    63   DA 0327    724         MTPR    (R3),@^PRIPL            ;FAULT IN CODE AND RAISE TO SYNCH
                                032C    725                                     ;ALL IN ONE INSTRUCTION
            00000000'GF   16 032C    726         JSB     G^MMG$INCPTREF          ;LOCK THE GLOBAL PAGE TABLE
              02 59    0D   E1 0332    727         BBC     #SEC$V_RESIDENT,R9,65$  ;NOT RESIDENT
                       29   10 0336    728         BSBB    100$                    ;CALCULATE A NEW R2 AMONG OTHER THINGS
                                0338    729 65$:    SETIPL  #IPL$_ASTDEL            ;BACK TO PAGE FAULTABLE STATE
                 83    52   D0 033B    730         MOVL    R2,(R3)+                ;STORE NEW GPT ENTRY
                    E3 57   F5 033E    731         SOBGTR  R7,60$                  ;LOOP THROUGH SPECIFIED NO. OF PTE'S
                       63   D4 0341    732         CLRL    (R3)                    ;ZERO STOPPER AT END OF GLOBAL SECTION
                 0088 8F   BA 0343    733         POPR    #^M<R3,R7>              ;RESTORE THE INITIAL SVAGPT AND COUNT
```

```
                03 59    0D   E1  0347   734          BBC     #SEC$V_RESIDENT,R9,70$  ;NOT RESIDENT
                      0B64   30  034B   735          BSBW    READ_RESIDENT           ;INITIALIZE THE CONTENTS
          50    59    01   0F  EF  034E   736 70$:     EXTZV   #SEC$V_SYSGBL,#1,R9,R0  ;SYSTEM OR GROUP GLOBAL BIT
     50   00000000'GF40    7E  0353   737          MOVAQ   G^EXE$GL_GSDGRPFL[R0],R0 ;ADDRESS OF APPROPRIATE LIST
                           035B   738
                           035B   739 ;
                           035B   740 ; QUEUE THE NEW GSD ON THE FRONT OF THE APPROPRIATE LIST.  PLACING IT ON THE
                           035B   741 ; FRONT ALLOWS FOR THE INSTALLATION OF A NEW COPY OF SAY A FORTRAN OTS WITH
                           035B   742 ; AN IDENT THAT IS GREATER THAN THE OLD BUT UPWARD COMPATIBLE.  THE NEW ONE
                           035B   743 ; IS FOUND FIRST IF THE MATCH IS MATLEQ.  THE OLD ONE IS STILL AVAILABLE FOR
                           035B   744 ; MATCH EQUAL.
                           035B   745 ;
                60    66   0E   035B   746          INSQUE  (R6),(R0)               ;PLACE GSD ON THE SYSTEM OR GROUP GLOBAL LIS
                      0151 31   035E   747          BRW     MAPGBLSEC2              ;GO MAP THE SECTION
                           0361   748
                           0361   749 ;
                           0361   750 ; SET UP ONE PAGE OF A RESIDENT GLOBAL SECTION
                           0361   751 ; ALLOCATE A PFN AND SET UP PFN DATA BASE
                           0361   752 ; CALCULATE THE NEW PTE CONTENTS
                           0361   753 ;
                   7E    52   7D   0361   754 100$:    MOVQ    R2,-(SP)
                00000000'EF    16  0364   755          JSB     MMG$ALLOCPFN            ;GET A PHYSICAL PAGE
                      52    8E   7D  036A   756          MOVQ    (SP)+,R2
                           50   D5  036D   757          TSTL    R0                      ;DID WE GET A PAGE
                           46   19  036F   758          BLSS    120$                    ;NO - GO WAIT
          52    15    00   50   F0  0371   759          INSV    R0,#PTE$V_PFN,#PTE$S_PFN,R2 ;ADD PFN TO PTE
           52   04400000   8F   CA  0376   760          BICL    #PTE$M_TYP0!PTE$M_TYP1,R2 ;CLEAR THE SECTION INDICATOR
                00 52      1F   E2  037D   761          BBSS    #PTE$V_VALID,R2,110$    ;SET THE VALID BIT TO INDICATE RESIDENT
     00000000'FF40    07   90   0381   762 110$:    MOVB    #PFN$C_ACTIVE,@PFN$AB_STATE[R0]
     00000000'FF40    02   90   0389   763          MOVB    #PFN$C_GLOBAL,@PFN$AB_TYPE[R0]
     00000000'FF40    53   D0   0391   764          MOVL    R3,@PFN$AL_PTE[R0]
     00000000'FF40    01   B0   0399   765          MOVW    #1,@PFN$AW_REFCNT[R0]
00000000'FF40    16   A6   3C   03A1   766          MOVZWL  GSD$W_GSTX(R6),@PFN$AL_BAK[R0]
00000000'FF40   00400000   8F   C8  03AA   767          BISL    #PTE$M_TYP0,@PFN$AL_BAK[R0] ;INDICATE SECTION BACKING STORE
                           05   03B6   768          RSB
                           03B7   769
          54   00000000'EF    D0  03B7   770 120$:    MOVL    SCH$GL_CURPCB,R4
          50   00000000'EF    7E  03BE   771          MOVAQ   SCH$GQ_FPGWQ,R0         ;WE WILL BE ON FREE PAGE WAIT QUEUE
               00000000'EF    16  03C5   772          JSB     MMG$PGFLTWAIT           ;PUT PROCESS ON THE QUEUE
                      7E      DC  03CB   773          MOVPSL  -(SP)                   ;CONSTRUCT PC-PSL TO RETURN HERE
               00000000'EF    16  03CD   774          JSB     MMG$SVPCTX              ;WE GET BACK BY REI
                      8C      11  03D3   775          BRB     100$                    ;GO TRY AGAIN
                           03D5   776
                           03D5   777 PRIPL:                                       ;THE PROCESSOR REGISTER NUMBER IS
                00000012   03D5   778          .LONG   PR$_IPL                     ;PLACED IN THIS LOCATION, INSTEAD OF
                           03D9   779                                              ;BEING REFERENCED SYMBOLICALLY ABOVE,
                           03D9   780                                              ;TO FAULT IN THE CODE THAT IS TO BE
                           03D9   781                                              ;EXECUTED AT RAISED IPL.
                           03D9   782          ASSUME  <.-60$> LE 512              ;PREVENT INTERVENING PAGE
                           03D9   783
                           03D9   784 ;
                           03D9   785 ; GLOBAL PAGE TABLE FULL
                           03D9   786 ; R0 = SYSTEM STATUS CODE
                           03D9   787 ; R5 = SYSTEM PROCESS HEADER ADDRESS
                           03D9   788 ; R10 = SECTION TABLE ENTRY ADDRESS
                           03D9   789 ;
                           03D9   790 GPTFULL:
```

```
                000000C4 8F    DD  03D9    791          PUSHL   #SS$_GPTFULL            ;GLOBAL PAGE TABLE FULL
                                   03DF    792 FRESTXERR:
                      0C AA    D5  03DF    793          TSTL    SEC$L_WINDOW(R10)
                         08    12  03E2    794          BNEQ    79$
00000000'EF             1C AA  C0  03E4    795          ADDL    SEC$L_PAGCNT(R10),MMG$GL_GBLPAGFIL ;PAGE FILE BACKING STORE - ADJUST
                         6A    D4  03EC    796 79$:     CLRL    SEC$L_GSD(R10)          ;GSD WILL BE DELETED BELOW
                               03EE    797          ASSUME SEC$L_PAGCNT EQ SEC$L_REFCNT+4 ;FIELDS MUST BE ADJACENT
                         18 AA 7C  03EE    798          CLRQ    SEC$L_REFCNT(R10)       ;SET INDICATORS FOR DELGBLSEC
                   00 14 AA    0E  E5  03F1    799          BBCC    #SEC$V_PERM,SEC$W_FLAGS(R10),80$ ;LET SECTION BE DELETED
                   00 36 A5    01  E6  03F6    800 80$:     BBSSI   #PHD$V_DALCSTX,PHD$W_FLAGS(R5),90$ ;SET DELETION FLAG
                     00000000'EF   16  03FB    801 90$:     JSB     MMG$DALCSTXSCN          ;SCAN FOR SECTIONS TO DEALLOCATE
                           50    8E  D0  0401    802          MOVL    (SP)+,R0                ;GET ERROR CODE
                                     0404    803 ;
                                     0404    804 ; GLOBAL SECTION TABLE FULL
                                     0404    805 ; R0 = SYSTEM STATUS CODE
                                     0404    806 ;
                                     0404    807 GSTFULL:
                               50    DD  0404    808          PUSHL   R0                      ;REMEMBER ERROR CODE TO RETURN
                            50 56    D0  0406    809          MOVL    R6,R0                   ;SET ADR OF BLOCK TO BE DEALLOCATED
                     00000000'GF    16  0409    810          JSB     G^EXE$DEAPAGED          ;RELEASE GLOBAL SECTION DESCRIPTOR
                               02    11  040F    811          BRB     ULKGSDMTXRET2           ;GO UNLOCK GSD MUTEX
                                     0411    812 ;
                                     0411    813 ; UNLOCK THE GLOBAL SECTION MUTEX AND RETURN WITH STATUS CODE IN R0
                                     0411    814 ;
                                     0411    815 ULKGSDMTXRET:
                               50    DD  0411    816          PUSHL   R0                      ;SAVE STATUS CODE
                                     0413    817 ULKGSDMTXRET2:
                        FBEA'  30    0413    818          BSBW    MMG$GSDMTXULK           ;UNLOCK THE GSD MUTEX
                                     0416    819                                          ;RETURNS R4=PCB ADR FOR MMG$DELGBLWCB
                        FBE7'  30    0416    820          BSBW    MMG$DELGBLWCB           ;DELETE ANY GLOBAL WINDOW CONTROL BLOCKS
                          50 8ED0    0419    821          POPL    R0                      ;RESTORE ERROR STAUS CODE TO RETURN
                                     041C    822          SETIPL  B^MMG$L_CALLEDIPL(FP)   ;RESTORE CALLER'S IPL
                                     0420    823 MAPSEC_RET:
                                  04 0420    824          RET
                                     0421    825
                                     0421    826          .DSABL  LSB
```

```
0421    828          .SBTTL  MGBLSC - MAP GLOBAL SECTION
0421    829 ;++
0421    830 ; FUNCTIONAL DESCRIPTION:
0421    831 ;
0421    832 ;       THE MAP GLOBAL SECTION SYSTEM SERVICE MAPS A SPECIFIED GLOBAL
0421    833 ; SECTION INTO A SPECIFIED RANGE OF VIRTUAL ADDRESS SPACE.
0421    834 ;
0421    835 ; CALLING SEQUENCE:
0421    836 ;
0421    837 ;       CALLG   ARGLIST,@#SYS$MGBLSC
0421    838 ;
0421    839 ; INPUT PARAMETERS:
0421    840 ;
0421    841 ;       INADR(AP) = ADDRESS OF 2 LONG WORDS THE 1ST OF WHICH SPECIFIES
0421    842 ;                   THE STARTING VIRTUAL ADDRESS TO CREATE, THE 2ND SPECIFIES
0421    843 ;                   THE ENDING VIRTUAL ADDRESS TO CREATE (INCLUSIVE).
0421    844 ;       RETADR(AP) = ADDRESS OF A 2 LONGWORD ARRAY INTO WHICH IS RETURNED
0421    845 ;                   THE STARTING AND ENDING VIRTUAL ADDRESSES (INCLUSIVE)
0421    846 ;                   OF THE PAGES JUST CREATED
0421    847 ;       ACMODE(AP) = THE ACCESS MODE (MAXIMIZED WITH CALLING MODE)
0421    848 ;                   USED AS THE OWNER OF THE NEW PAGE(S)
0421    849 ;       FLAGS(AP)  = BIT  0  - GBL - GLOBAL IF SET, PROCESS IF CLEAR
0421    850 ;                    BIT  1  - CRF - COPY ON REFERENCE
0421    851 ;                    BIT  2  - DZRO - DEMAND ZERO
0421    852 ;                    BIT  3  - WRT - WRITABLE IF SET, READ ONLY IF CLEAR
0421    853 ;                    BITS 4 - 5, RESERVED, MUST BE ZERO
0421    854 ;                    BITS 6 & 7 - WRTMOD - WRITE ACCESS FOR SECTION
0421    855 ;                    BITS 8 - 13 RESERVED, MUST BE ZERO
0421    856 ;                    BIT  14 - PERM - PERMANENT IF SET, TEMPORARY IF CLEAR
0421    857 ;                    BIT  15 - SYSGBL - SYSTEM GLOBAL IF SET, GROUP GLOBAL IF CLEAR
0421    858 ;                    BIT  16  - PFNMAP - MAP TO SPECIFIC PFN'S, IF SET
0421    859 ;                    BIT  17  - EXPREG - MAP TO FIRST FREE SPACE AVAILABLE
0421    860 ;                    BIT  18 - PROTECT - SET IF WRITE ACCESS MODE SPECIFIED
0421    861 ;                    BITS 19  - 31 RESERVED, MUST BE ZERO
0421    862 ;           GSDNAM(AP) = THE DESCRIPTOR OF THE GLOBAL SECTION NAME
0421    863 ;           IDENT(AP)  = ADDRESS OF QUAD WORD CONTAINING SECTION IDENTIFICATION
0421    864 ;                    FIRST LONG WORD CONTAINS THE MATCH CONTROL INFORMATION
0421    865 ;                       0 = ISD$K_MATALL, MATCH ALWAYS
0421    866 ;                       1 = ISD$K_MATEQU, MATCH IF IDENTS ARE EQUAL
0421    867 ;                       2 = ISD$K_MATLEQ, MATCH IF HIGH 8 BITS ARE EQUAL
0421    868 ;                           AND LOW 24 BITS ARE LESS THAN OR EQUAL TO
0421    869 ;                           THE ID STORED IN THE GLOBAL SECTION.
0421    870 ;                    SECOND LONG WORD CONTAINS THE IDENT TO BE COMPARED
0421    871 ;           RELPAG(AP) = RELATIVE PAGE IN SECTION TO START MAPPING
0421    872 ;
0421    873 ; IMPLICIT INPUTS:
0421    874 ;
0421    875 ;       NONE
0421    876 ;
0421    877 ; OUTPUT PARAMETERS:
0421    878 ;
0421    879 ;       NONE
0421    880 ;
0421    881 ; IMPLICIT OUTPUTS:
0421    882 ;
0421    883 ;       NONE
0421    884 ;
```

```
0421    885 ; COMPLETION CODES:
0421    886 ;
0421    887 ;       SS$_NORMAL - SUCCESS
0421    888 ;       SS$_IVSECFLG - INVALID SECTION FLAGS
0421    889 ;       SS$_ENDOFFILE - END OF FILE ENCOUNTERED
0421    890 ;       SS$_NOPRIV - NO PRIVILEGE FOR ATTEMPTED OPERATION
0421    891 ;       SS$_VASFULL - VIRTUAL ADDRESS SPACE FULL
0421    892 ;       SS$_IVLOGNAM - INVALID LOGICAL NAME
0421    893 ;       SS$_TOOMANYLNAM - TOO MANY LOGICAL NAMES (DEPTH > 10)
0421    894 ;       SS$_SHMNOTCNCT - SHARED MEMORY DATA STRUCTURES NOT CONNECTED
0421    895 ;       SS$_NOSUCHSEC - NO SUCH GLOBAL SECTION
0421    896 ;       SS$_PAGOWNVIO - PAGE OWNER VIOLATION (RE-MAPPING PAGES ALREADY IN USE)
0421    897 ;       SS$_EXQUOTA - EXCEEDED PAGING FILE QUOTA (FOR CRF SECTIONS)
0421    898 ;
0421    899 ; SIDE EFFECTS:
0421    900 ;
0421    901 ;       NONE
0421    902 ;
0421    903 ;--
```

```
                                0421  ;
                                0421  ; ***************************************************************
                                0421  ;
                                0421  ; **************** THE FOLLOWING CODE MAY BE PAGED *****************
                                0421  ;
                            00000421          .PSECT  YF$$SYSCRMPSC
                                0421
                                0421  INADRERR:
                            04  0421          RET
                                0422  ;
                                0422  ; ***************************************************************
                                0422  ;
                                0422
                            00000008          .PSECT  Y$EXEPAGED              ;PUT ENTRY POINT INTO SEPARATE PSECT
                                0008
                                0008  ;*************  EXE$MGBLSC  ENTRY POINT
                                0008
                          0FFC  0008          .ENTRY  EXE$MGBLSC,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          00000422'EF       17  000A          JMP     EXE_MGBLSC              ;TRANSFER TO REAL PROCEDURE
                                0010
                            00000422          .PSECT  YF$$SYSCRMPSC           ;BACK TO $CRMPSC PSECT
                                0422
                                0422  EXE_MGBLSC:
                5E    1C    C2  0422          SUBL    S^#-MMG$C_LENGTH,SP     ;RESERVE A SCRATCH STORAGE AREA
          00000000'GF       16  0425          JSB     G^MMG$INADRINI          ;GET INPUT RANGE TO R4, R5
                                042B                                          ;INIT RETURN RANGE AND SCRATCH AREA
                F3 50       E9  042B          BLBC    R0,INADRERR
                   00       DD  042E          PUSHL   #0                      ;REMEMBER NO START RETADR YET
                   01       DD  0430          PUSHL   #SS$_NORMAL             ;SAVE SUCCESS RETURN CODE
                   30       BB  0432          PUSHR   #^M<R4,R5>              ;PUSH ENDVA, STARTVA
            50   10 AC      D0  0434          MOVL    FLAGS(AP),R0            ;SECTION FLAGS
                 FBC5'      30  0438          BSBW    MMG$VFYSECFLG           ;VERIFY SECTION FLAGS
                59   50     D0  043B          MOVL    R0,R9                   ;USE VERIFIED SECTION FLAGS
      54    00000000'GF     D0  043E          MOVL    G^SCH$GL_CURPCB,R4      ;PROCESS CONTROL BLOCK ADDRESS
      50    00000000'GF     DE  0445          MOVAL   G^EXE$GL_GSDMTX,R0      ;GLOBAL SECTION MUTEX
          00000000'GF       16  044C          JSB     G^SCH$LOCKW             ;LOCK GSD FOR WRITING
          00000000'EF       16  0452          JSB     MMG$DALCSTXSCN1         ;SCAN FOR SECTIONS TO DELETE
                                0458  ;
                                0458  ; RETURNS AT ASTDEL
                                0458  ;
                5E   34     C2  0458          SUBL    #<11*4>+8,SP            ;RESERVE SPACE FOR GSD NAME AND IDENT
                57   5E     D0  045B          MOVL    SP,R7                   ;ADDRESS OF SCRATCH SPACE
                                045E
                                045E          ASSUME  IDENT EQ GSDNAM+4
            50   14 AC      7D  045E          MOVQ    GSDNAM(AP),R0           ;DESCRIPTOR ADDRESS FOR GLOBAL SECTION NAME
                                0462                                          ;ADDRESS OF IDENTIFICATION QUAD WORD
                56   59     D0  0462          MOVL    R9,R6                   ;SECTION FLAGS
                 FB98'      30  0465          BSBW    MMG$GSDSCN              ;SCAN GSD'S FOR THIS NAME
                0C 50       E8  0468          BLBS    R0,5$                   ;BRANCH IF SECTION FOUND
      50    00000978 8F     D1  046B          CMPL    #SS$_NOSUCHSEC,R0       ;WAS THE ERROR DUE TO SECTION NOT FOUND?
                   9D       12  0472          BNEQ    ULKGSDMTXRET            ;IF NO, RETURN ERROR CODE TO CALLER
                                0474  ;
                                0474  ;  0(SP) = ASCIC GLOBAL SECTION NAME STRING
                                0474  ; 44(SP) = IDENT INFORMATION
                                0474  ; 52(SP) = STARTVA
                                0474  ; 56(SP) = ENDVA
                                0474  ; 60(SP) = SUCCESS CODE (SS$_NORMAL OR SS$_EXISTED) TO BE RETURNED IF
```

```
                                0474   962 ;                   MAPPING IS SUCCESSFUL.
                                0474   963 ; R4 = SHARED MEMORY CONTROL BLOCK ADDRESS IF GLOBAL SECTION IS IN SH MEM
                                0474   964 ; R5 = SYSTEM PROCESS HEADER ADDRESS
                                0474   965 ; R6 = GLOBAL SECTION DESCRIPTOR BLOCK ADDRESS
                                0474   966 ; R9 = SECTION FLAGS
                                0474   967 ; R10 IS   0 IF THE GSD WAS FOUND IN LOCAL MEMORY
                                0474   968 ;          -1 IF THE LOCAL MEMORY SEARCH EXTENDED INTO SHARED MEMORY TABLES
                                0474   969 ;          >0 IF A SPECIFIC SHARED MEMORY NAME WAS SPECIFIED
                                0474   970 ;
                                0474   971 ;
                                0474   972 ; IF THE GLOBAL SECTION WAS NOT FOUND, PERHAPS IT IS BECAUSE THE SECTION IS
                                0474   973 ; COPY AND REFERENCE AND THERE WAS A LOGICAL NAME WHICH STATED IT WAS IN
                                0474   974 ; SHARED MEMORY.  SINCE CRF SECTIONS ARE PLACED IN LOCAL MEMORY INSTEAD OF
                                0474   975 ; SHARED MEMORY, USE THE RESULTANT GLOBAL SECTION NAME AND SEARCH FOR THAT
                                0474   976 ; IN LOCAL MEMORY.  THIS IS DONE BY FORCING NO LOGICAL TRANSLATION OF THE
                                0474   977 ; GLOBAL SECTION NAME (PREFIXING IT WITH AN UNDERSCORE).
                                0474   978 ;
                    FB89'  30   0474   979          BSBW    MMG$FINDGSNOTRN         ;FIND GSD WITH NO LOGICAL TRANSLATION
                 5E    34  C0   0477   980 5$:      ADDL2   #<11*4>+8,SP            ;RECLAIM THE SCRATCH AREA
                   94  50  E9   047A   981          BLBC    R0,ULKGSDMTXRET         ;BRANCH IF SECTION NOT FOUND
                                047D   982 ;
                                047D   983 MAPGBLSEC1:
                29   0A A6  91   047D   984          CMPB    GSD$B_TYPE(R6),#DYN$C_SHMGSD ;IS GS IN SHARED MEMORY?
                     0F     13   0481   985          BEQL    7$                      ;BR ON YES, CAN'T ALWAYS ACCESS SEC TBL
                5A   16 A6  32   0483   986          CVTWL   GSD$W_GSTX(R6),R10      ;IS GS MAPPED TO A FILE?
                     09     13   0487   987          BEQL    7$                      ;BR ON NOT MAPPED TO A FILE
          50    55   20 A5  C1   0489   988          ADDL3   PHD$L_PSTBASOFF(R5),R5,R0 ;BASE ADDRESS OF SECTION TABLE
                5A  604A    DE   048E   989          MOVAL   (R0)[R10],R10           ;ADDRESS OF SECTION TABLE ENTRY
          50    20 A6   1E   9C   0492   990 7$:      ROTL    #31-SEC$V_CRF,GSD$W_FLAGS(R6),R0 ;GET FLAGS FROM GSD
                     05     18   0497   991          BGEQ    10$                     ;BRANCH IF NOT COPY ON REFERENCE
             00 FC AD   08   E2   0499   992          BBSS    S^#MMG$V_CHGPAGFIL,B^MMG$L_MAXACMODE(FP),10$
                                049E   993                                          ;MUST CHARGE PAGES AGAINST PAGE FILE QUOTA
                     06     EF   049E   994 10$:     EXTZV   #<SEC$V_AMOD-<SEC$V_CRF+1>>,- ;GET ACCESS MODE OF SECTION
                50   50     02   04A0   995                  #SEC$S_AMOD,R0,R0
                  50   FC AD  91   04A3   996          CMPB    B^MMG$L_MAXACMODE(FP),R0 ;SEE IF THIS CALLER CAN MAP THE SECTION
                     42     14   04A7   997          BGTR    MGBLSC_NOPRIV           ;BRANCH IF CANNOT MAP THE SECTION
                15   0A A6  91   04A9   998          CMPB    GSD$B_TYPE(R6),#DYN$C_GSD ;WHAT TYPE OF GSD IS THIS?
                     03     12   04AD   999          BNEQU   MAPGBLSEC2              ;BR IF NOT LOCAL MEMORY TYPE
                   18 AA    D6   04AF  1000          INCL    SEC$L_REFCNT(R10)       ;BIAS THE SECTION REFERENCE COUNT
                                04B2  1001                                          ;SO THAT IT CAN'T BE DELETED
                                04B2  1002 ;
                                04B2  1003 ; R5  = SYSTEM PROCESS HEADER ADDRESS
                                04B2  1004 ; R9  = SECTION FLAGS
                                04B2  1005 ; R10 = SECTION TABLE ENTRY ADDRESS
                                04B2  1006 ;
                                04B2  1007          ASSUME DYN$C_EXTGSD GT DYN$C_GSD
                                04B2  1008          ASSUME DYN$C_EXTGSD LT DYN$C_SHMGSD
                                04B2  1009 MAPGBLSEC2:
                28   0A A6  91   04B2  1010          CMPB    GSD$B_TYPE(R6),#DYN$C_EXTGSD ;IS THIS EXTENDED GSD?
                     03     12   04B6  1011          BNEQU   MAPGBLSEC3              ;BR IF LOCAL MEMORY OR SHARED MEMORY
                   2C A6    D6   04B8  1012          INCL    GSD$L_REFCNT(R6)        ;YES, BIAS GSD REFCNT SO IT CAN'T
                                04BB  1013                                          ;BE DELETED BUT MUTEX CAN BE UNLOCKED
                                04BB  1014 MAPGBLSEC3:
                                04BB  1015          .ENABL  LSB
                     54     DD   04BB  1016          PUSHL   R4                      ;SAVE SHB ADDRESS
                  FB40'     30   04BD  1017          BSBW    MMG$GSDMTXULK           ;UNLOCK THE GLOBAL SECTION MUTEX
                    54 8ED0    04C0  1018          POPL    R4                      ;RESTORE SHB ADDRESS
```

```
                5B   56   D0  04C3  1019          MOVL    R6,R11                          ;SAVE GSD ADR FOR ERROR PATH RECOVERY
              65 59   11   E1  04C6  1020          BBC     #SEC$V_EXPREG,R9,MAPGBLSEC4 ;BR IF RANGE IS EXPLICITLY STATED
                                04CA  1021 ;
                                04CA  1022 ; FIND THE FIRST AVAILABLE VIRTUAL ADDRESS AND COMPUTE THE RANGE TO BE MAPPED.
                                04CA  1023 ;
              28    0A A6   91  04CA  1024          CMPB    GSD$B_TYPE(R6),#DYN$C_EXTGSD ;WHAT TYPE OF GSD IS THIS?
                       15   1F  04CE  1025          BLSSU   30$                             ;BR IF LOCAL MEMORY GSD
                       1C   13  04D0  1026          BEQL    40$                             ;BR IF EXTENDED GSD
                       51   D4  04D2  1027          CLRL    R1                              ;INITIALIZE COUNT OF PAGES IN SHMGSD
                    57    04   9A  04D4  1028          MOVZBL  #GSD$C_PFNBASMAX,R7             ;MAX NUMBER OF BASES ALLOWED
              58     54 A6   9E  04D7  1029          MOVAB   GSD$L_BASPFN1(R6),R8            ;ADR OF FIRST GSD BASE PFN
                       88   D5  04DB  1030 10$:      TSTL    (R8)+                           ;IS THIS BASE USED?
                    51    88   C0  04DD  1031          ADDL2   (R8)+,R1                        ;ADD THESE PAGES INTO SIZE OF SECTION
                       F8 57   F5  04E0  1032 20$:      SOBGTR  R7,10$                          ;REPEAT FOR NEXT BASE
                       0D   11  04E3  1033          BRB     50$                             ;JOIN COMMON CODE
              51     1C AA   D0  04E5  1034 30$:      MOVL    SEC$L_PAGCNT(R10),R1            ;SIZE OF LOCAL MEMORY GLOBAL SECTION
                       07   11  04E9  1035          BRB     50$                             ;JOIN COMMON CODE
                                04EB  1036
                                04EB  1037 MGBLSC_NOPRIV:
                      FBF5   31  04EB  1038          BRW     CRMPSC_NOPRIV                   ;NO PRIV TO MAP GLOBAL SECTION
                                04EE  1039
              51     28 A6   D0  04EE  1040 40$:      MOVL    GSD$L_PAGES(R6),R1              ;SIZE OF EXTENDED GSD SECTION
              50     1C AC   D0  04F2  1041 50$:      MOVL    RELPAG(AP),R0                   ;GET # OF PAGES NOT TO BE MAPPED
                       2A   19  04F6  1042          BLSS    55$                             ;BR IF ILLEGAL NUMBER OF PAGES
                    51    50   D1  04F8  1043          CMPL    R0,R1                           ;ARE MORE PAGES SKIPPED THAN IN SECTION?
                       25   18  04FB  1044          BGEQ    55$                             ;BR IF ILLEGAL NUMBER OF PAGES
                    51    50   C2  04FD  1045          SUBL    R0,R1                           ;GET # OF PAGES IN SECTION TO MAP
                 51 51    09   78  0500  1046          ASHL    #9,R1,R1                        ;CONVERT PAGE COUNT TO # OF
                       51   D7  0504  1047          DECL    R1                              ;BYTES BETWEEN START VA AND END VA
              50    00000000'9F   D0  0506  1048          MOVL    @#CTL$GL_PHD,R0                 ;GET PROCESS HEADER FOR PROCESS
                       0C   BA  050D  1049          POPR    #^M<R2,R3>                      ;FIND REGION TO MAP SECTION INTO
                    12 52   1E   E1  050F  1050          BBC     #VA$V_P1,R2,60$                 ;BR IF MAPPING INTO P0 SPACE
53    30 A0   000001FF 8F   C1  0513  1051          ADDL3   #^X1FF,PHD$L_FREP1VA(R0),R3 ;ENDING VA IN P1 SPACE
                 52    53   51  C3  051C  1052          SUBL3   R1,R3,R2                        ;STARTING VA IN P1 SPACE
                       0B   11  0520  1053          BRB     70$                             ;JOIN COMMON CODE
                    00EE   31  0522  1054 55$:      BRW     ILL_RELPAG                      ;BR TO RETURN ERROR CODE
                 52    28 A0   D0  0525  1055 60$:      MOVL    PHD$L_FREP0VA(R0),R2            ;STARTING VA IN P0 SPACE
                 53    52   51  C1  0529  1056          ADDL3   R1,R2,R3                        ;ENDING VA IN P0 SPACE
                       0C   BB  052D  1057 70$:      PUSHR   #^M<R2,R3>                      ;REMEMBER VA RANGE TO BE MAPPED
                                052F  1058          .DSABL  LSB
                                052F  1059 ;
                                052F  1060 ; BUILD A PAGE TABLE ENTRY TO PUT IN THE PROCESS PAGE TABLE
                                052F  1061 ;
                                052F  1062 MAPGBLSEC4:
                                052F  1063          .ENABLE LSB
              28    0A A6   91  052F  1064          CMPB    GSD$B_TYPE(R6),#DYN$C_EXTGSD ;IS THIS A MAP BY PFN?
                       3E   1F  0533  1065          BLSSU   6$                              ;NO, BR TO CREATE LOCAL MEM GPTX PTE
                       2D   13  0535  1066          BEQL    5$                              ;YES, BR TO CREATE VALID WINDOW PTE
                    57    04   9A  0537  1067          MOVZBL  #GSD$C_PFNBASMAX,R7             ;GET # OF BASES ALLOWED
              5A     54 A6   9E  053A  1068          MOVAB   GSD$L_BASPFN1(R6),R10           ;GET ADR OF FIRST PFN BASE
                                053E  1069 MAP_NXT_BASE:
                    58    8A   D0  053E  1070          MOVL    (R10)+,R8                       ;GET PFN BASE FOR FIRST PIECE OF SECTION
                    59    8A   D0  0541  1071          MOVL    (R10)+,R9                       ;GET SIZE OF FIRST PIECE OF SECTION
                       03   12  0544  1072          BNEQ    3$                              ;BR IF MORE SECTION TO MAP
                       023B 31  0546  1073          BRW     NO_MORE_PAGES                   ;BR IF ALL SECTION MAPPED
              58    10 A4   C0  0549  1074 3$:       ADDL2   SHB$L_BASGSPFN(R4),R8           ;ADD IN BASE PFN FOR GS PAGES
              00 58   1F   E3  054D  1075          BBCS    #PTE$V_VALID,R8,4$              ;SET VALID BIT IN PTE
```

```
                                                                L 2
                0600 8F    BB  0551  1076 4$:     PUSHR   #^M<R9,R10>             ;SAVE ADR OF PFN BASES AND PIECE SIZE
                     5A    D4  0555  1077          CLRL    R10                     ;FORCE USE OF GSD FLAGS
             59    E8 AD    D0  0557  1078          MOVL    MMG$L_VFYFLAGS(FP),R9   ;RESTORE FLAGS FOR SETSECPROTOWN CALL
                   0438     30  055B  1079          BSBW    SETSECPROTOWN           ;SET SECTION PROTECTION AND OWNER
                0600 8F    BA  055E  1080          POPR    #^M<R9,R10>             ;RESTORE ADR OF PFN BASES AND PIECE SIZE
                     18    11  0562  1081          BRB     8$                      ;BR PAST NO SECTION TABLE CODE
                     5A    D4  0564  1082 5$:     CLRL    R10                     ;REMEMBER NO SECTION TABLE ENTRY EXISTS
                58   24 A6    D0  0566  1083          MOVL    GSD$L_BASEPFN(R6),R8    ;CREATING VALID PTE
          58    80200000 8F    C8  056A  1084          BISL2   #<PTE$M_VALID ! PTE$M_WINDOW>,R8 ;SET VALID & WINDOW
                     06    11  0571  1085          BRB     7$                      ;SKIP PAGE TABLE INDEX
                16   00    EF  0573  1086 6$:     EXTZV   #PTE$V_GPTX,#PTE$S_GPTX,- ;GET THE FIRST GLOBAL
                58   08 AA        0576  1087                  SEC$L_GPXPFC(R10),R8    ;PAGE TABLE INDEX
                   041A     30  0579  1088 7$:     BSBW    SETSECPROTOWN           ;SET SECTION PROTECTION AND OWNER
                5B   56    D0  057C  1089 8$:     MOVL    R6,R11                  ;SAVE GSD ADDRESS FOR MMG$MAPSECPAG
                  67 50    E9  057F  1090          BLBC    R0,13$                  ;BRANCH IF PROTECTION VIOLATION
                50   01    D1  0582  1091          CMPL    #SS$_NORMAL,R0          ;WAS THERE AN ALTERNATE SUCCESS CODE?
                     04    13  0585  1092          BEQL    9$                      ;BR IF CODE WAS NORMAL SUCCESS
             08 AE   50    D0  0587  1093          MOVL    R0,8(SP)                ;SET ALTERNATE RETURN CODE
                             058B  1094
                             058B  1095 ; BECAUSE OF THE REGISTERS NEEDED BY EXE$CHECKPROT_16 FOR INPUT, SAVE
                             058B  1096 ; R2 THROUGH R6 NOW.
                             058B  1097
                0070 8F    BB  058B  1098 9$:     PUSHR   #^M<R2,R3,R4,R5,R6>     ;SAVE SOME WORK REGISTERS
                             058F  1099 ;
                             058F  1100 ; R4= SHB ADDRESS, IF GLOBAL SECTION IS IN SHARED MEMORY
                             058F  1101 ; VALIDATE THIS PROCESS' ACCESS TO THIS FILE
                             058F  1102 ; NOTE: SINCE REFCNT IS INCREMENTED ABOVE, GSD CAN'T GO AWAY EVEN THOUGH
                             058F  1103 ; GSD MUTEX IS RELEASED.
                             058F  1104 ;
             55   14 A6    9E  058F  1105          MOVAB   GSD$W_PROT(R6),R5       ;SET PROTECTION WORD ADDRESS
             7E   10 A6    D0  0593  1106          MOVL    GSD$L_FILUIC(R6),-(SP)  ;GET FILE OWNER
                     04    12  0597  1107          BNEQ    10$                     ;USE THE FILE UIC
             6E   0C A6    D0  0599  1108          MOVL    GSD$L_PCBUIC(R6),(SP)   ;NO FILE - USE SECTION CREATOR UIC
                             059D  1109 ;
                             059D  1110 ; CHECK THAT THE CALLER HAS READ OR WRITE ACCESS TO THE SECTION
                             059D  1111 ; ACCORDING TO HOW IT IS TO BE MAPPED.
                             059D  1112 ;
          50    00000000'GF    D0  059D  1113 10$:    MOVL    G^SCH$GL_CURPCB,R0      ;SET CURRENT PROCESS CONTROL BLOCK ADR
             50   008C C0    D0  05A4  1114          MOVL    PCB$L_ARB(R0),R0        ;GET ACCESS RIGHTS BLOCK
                54   20 A0    9E  05A9  1115          MOVAB   ARB$L_RIGHTSLIST(R0),R4 ;SET RIGHTSLIST ADDRESS
                   53   01    D0  05AD  1116          MOVL    #ARM$M_READ,R3          ;SET UP FOR READ ACCESS CHECK
             09 E8 AD    14    E1  05B0  1117          BBC     #SEC$V_EXECUTE,MMG$L_VFYFLAGS(FP),11$
                     59    DC  05B5  1118          MOVPSL  R9
                03 59    17    E0  05B7  1119          BBS     #PSL$V_PRVMOD+1,R9,11$  ;MAKE SURE PREVIOUS MODE WAS EXEC, KERNEL
                   53   04    D0  05BB  1120          MOVL    #ARM$M_EXECUTE,R3       ;SET UP FOR EXECUTE ACCESS CHECK
             13 E8 AD    03    E1  05BE  1121 11$:    BBC     #SEC$V_WRT,MMG$L_VFYFLAGS(FP),12$ ;BR IF NOT WRITING THE SECTION
                15   0A A6    91  05C3  1122          CMPB    GSD$B_TYPE(R6),#DYN$C_GSD ;IS IT LOCAL MEMORY GSD
                     0A    12  05C7  1123          BNEQ    111$                    ;NO - NEED WRITE ACCESS
             08 14 AA    01    E0  05C9  1124          BBS     #SEC$V_CRF,SEC$W_FLAGS(R10),12$ ;CRF DOESNT NEED WRITE ACCESS
             19 E8 AD    01    E0  05CE  1125          BBS     #SEC$V_CRF,MMG$L_VFYFLAGS(FP),14$ ;CALLER CAN'T GET CRF
                   53   02    D0  05D3  1126 111$:   MOVL    #ARM$M_WRITE,R3         ;SET UP FOR WRITE ACCESS CHECK
                   56   8E    D0  05D6  1127 12$:    MOVL    (SP)+,R6                ;GET OBJECT'S OWNER UIC
                   52   60    9E  05D9  1128          MOVAB   ARB$Q_PRIV(R0),R2       ;SET PRIV QUADWORD ADDRESS
                00000000'EF    16  05DC  1129          JSB     EXE$CHECKPROT_16        ;DO SOGW PROTECTION CHECK
                   0070 8F    BA  05E2  1130          POPR    #^M<R2,R3,R4,R5,R6>     ;RESTORE REGISTERS
                     0A 50    E8  05E6  1131          BLBS    R0,15$                  ;BRANCH IF CALLER HAS DESIRED ACCESS
                      0135    31  05E9  1132 13$:    BRW     PRE_MAP_ERR             ;BRANCH IF ACCESS DENIED
```

```
                     05EC  1133 ;
          50  016C 8F   3C  05EC  1134 14$:    MOVZWL  #SS$_IVSECFLG,R0        ;INVALID FLAGS SPECIFIED
                 F6   11  05F1  1135          BRB     13$
                      05F3  1136 ;
                      05F3  1137 ; R7 - COUNT OF PFN BASES ALLOWED IN GSD
                      05F3  1138 ; R10 - ADR OF NEXT PFN BASE IN GSD
                      05F3  1139 ;
          28   0A A6   91  05F3  1140 15$:    CMPB    GSD$B_TYPE(R6),#DYN$C_EXTGSD ;IS THIS AN EXTENDED GSD?
                    3B   1F  05F7  1141          BLSSU   MAP_LOCAL_MEM           ;NO, GO USE SECTION TABLE ENTRY
                    06   1A  05F9  1142          BGTRU   SHM_GSD                 ;NO, GO USE SHARED MEMORY GSD
          59   28 A6   D0  05FB  1143          MOVL    GSD$L_PAGES(R6),R9      ;YES, GET # OF PAGES IN SECTION FROM GSD
                    37   11  05FF  1144          BRB     30$                     ;SKIP FINDING # PAGES FROM SECTION TBL
                      0601  1145 SHM_GSD:
             57   04   91  0601  1146          CMPB    #GSD$C_PFNBASMAX,R7     ;IS THIS THE FIRST BASE?
                    4D   12  0604  1147          BNEQ    COMMON_MAP              ;NO, BR THEN TO SKIP RELPAG OFFSETING
          50   1C AC   D0  0606  1148          MOVL    RELPAG(AP),R0           ;GET STARTING RELATIVE PAGE TO MAP
               59   50   D1  060A  1149          CMPL    R0,R9                   ;SEE IF RELPAG IS PAST THIS PIECE
                    34   1F  060D  1150          BLSSU   32$                     ;BR IF MUST MAP PART OF THIS PIECE
                    57   D7  060F  1151 20$:     DECL    R7                      ;DEC COUNT OF PIECES OF SECTION TO MAP
                    08   12  0611  1152          BNEQ    TRY_NXT_BASE            ;BR IF ANOTHER PIECE TO MAP (MAYBE)
                      0613  1153 ILL_RELPAG:
          50  0870 8F   3C  0613  1154          MOVZWL  #SS$_ENDOFFILE,R0       ;REMEMBER THERE WERE NO PAGES TO MAP
                 0106   31  0618  1155          BRW     PRE_MAP_ERR             ;BR IF NO SECTION TO MAP AT ALL
                      061B  1156 TRY_NXT_BASE:
             58   8A   D0  061B  1157          MOVL    (R10)+,R8               ;GET PFN BASE OF NEXT PIECE
             59   8A   C0  061E  1158          ADDL2   (R10)+,R9               ;ADD NEXT BASE COUNT TO RELPAG OFFSET
             59   50   D1  0621  1159          CMPL    R0,R9                   ;IS RELPAG PAST THIS PIECE?
                  E9   1E  0624  1160          BGEQU   20$                     ;BR IF NOTHING IN THIS PIECE TO MAP
             59   50   C2  0626  1161          SUBL2   R0,R9                   ;GET OFFSET TO FIRST PFN TO MAP
     50   FC AA   59   C3  0629  1162          SUBL3   R9,-4(R10),R0           ;GET # OF PAGES IN PIECE NOT MAPPED
             58   50   C0  062E  1163          ADDL2   R0,R8                   ;GET FIRST PFN TO BE MAPPED
                FF15   31  0631  1164          BRW     3$                      ;BR TO CREATE PTE
                      0634  1165 MAP_LOCAL_MEM:
     59   1C AA   D0  0634  1166          MOVL    SEC$L_PAGCNT(R10),R9    ;NUMBER OF PAGES IN THE SECTION
     50   1C AC   D0  0638  1167 30$:     MOVL    RELPAG(AP),R0           ;STARTING RELATIVE PAGE TO MAP
               D5   19  063C  1168          BLSS    ILL_RELPAG              ;BR IF ILLEGAL RELPAG, NEGATIVE
          59   50   D1  063E  1169          CMPL    R0,R9                   ;IS RELPAG WITHIN THE SECTION?
               D0   18  0641  1170          BGEQ    ILL_RELPAG              ;BR IF ILLEGAL RELPAG, PAST SECTION
          59   50   C2  0643  1171 32$:     SUBL    R0,R9                   ;FEWER PAGES TO MAP
          58   50   C0  0646  1172          ADDL    R0,R8                   ;BIAS STARTING GPTX
     15   0A A6   91  0649  1173 35$:     CMPB    GSD$B_TYPE(R6),#DYN$C_GSD ;IS THIS NORMAL GSD?
               04   12  064D  1174          BNEQ    COMMON_MAP              ;BR ON NO, DON'T SET TYPO BIT
     00 58   16   E2  064F  1175          BBSS    #PTE$V_TYPO,R8,COMMON_MAP ;GLOBAL PAGE TYPE
                      0653  1176 ;
                      0653  1177 ; 0(SP) = STARVA, 4(SP) = ENDVA
                      0653  1178 ; R6 = GLOBAL SECTION DESCRIPTOR BLOCK ADDRESS
                      0653  1179 ; R8 = STARTING PTE CONTENTS, R9 = MAXIMUM PAGE COUNT TO MAP
                      0653  1180 ; R10 = GLOBAL SECTION TABLE ADDRESS
                      0653  1181 ;
                      0653  1182 COMMON_MAP:
               0C   BA  0653  1183          POPR    #^M<R2,R3>              ;R2 = STARTVA, R3 = ENDVA
          50   52   D2  0655  1184          MCOML   R2,R0                   ;STARTVA = -1?
               03   12  0658  1185          BNEQ    40$
             00A2   31  065A  1186          BRW     84$                     ;BRANCH IF YES, NO RANGE TO MAP
     15   0A A6   91  065D  1187 40$:     CMPB    GSD$B_TYPE(R6),#DYN$C_GSD ;IS THIS NORMAL GSD?
               2A   12  0661  1188          BNEQ    48$                     ;NO - DON'T TRY ANY OPTIMIZATIONS
          54   52   7D  0663  1189          MOVQ    R2,R4                   ;GET THE START AND END VA
```

```
                                                N 2
                00000000'EF   16  0666  1190          JSB     MMG$IN_REGION           ;IS IT NEW ADDRESS SPACE
                      1E 50   E9  066C  1191          BLBC    R0,48$                  ;NO - CREATE IT THE HARD WAY
                   59   56   D1  066F  1192          CMPL    R6,R9                   ;CHECK FIT OF SECTION IN THE SPACE
                        19   12  0672  1193          BNEQ    48$                     ;NOT EXACT - DO IT THE HARD WAY
             54 00000000'EF   D0  0674  1194          MOVL    SCH$GL_CURPCB,R4
                00000000'EF   16  067B  1195          JSB     MMG$TRY_ALL             ;SEE IF REGION CAN BE EXPANDED
                      09 50   E9  0681  1196          BLBC    R0,48$                  ;NO - CREATE IT THE HARD WAY
                       01ED   30  0684  1197          BSBW    FAST_MAP                ;DO IT THE FAST WAY
                   0498 8F    BB  0687  1198          PUSHR   #^M<R3,R4,R7,R10>       ;NO EFFECT - MATCHES A LATER POPR
                        2E    11  068B  1199          BRB     66$
                        59    D7  068D  1200 48$:     DECL    R9                      ;PAGE COUNT BASE 0
                   53   52    D1  068F  1201          CMPL    R2,R3                   ;CHECK DIRECTION OF MAPPING
                        09    1F  0692  1202          BLSSU   60$                     ;BRANCH IF FORWARDS
                        04    1A  0694  1203          BGTRU   50$                     ;BRANCH IF BACKWARDS
                03 52   1E    E1  0696  1204          BBC     #VA$V_P1,R2,60$         ;WHEN EQUAL, FORWARD IF P0, BACKWARDS IF P1
                   58   59    C0  069A  1205 50$:     ADDL    R9,R8                   ;START AT LAST GPTX WHEN MAPPING BACKWARDS
             56    0CE1'CF    DE  069D  1206 60$:     MOVAL   W^MMG$MAPSECPAG,R6      ;MAP SECTION PAGE ROUTINE
                   0498 8F    BB  06A2  1207          PUSHR   #^M<R3,R4,R7,R10>       ;SAV VA RANGE,SHB,PFNBASCNT,PFNBAS ADR
                29   0A AB    91  06A6  1208          CMPB    GSD$B_TYPE(R11),#DYN$C_SHMGSD ;IS THIS A SHARED MEMORY GSD?
                        02    12  06AA  1209          BNEQ    65$                     ;BR IF IT IS NOT
                        5A    D4  06AC  1210          CLRL    R10                     ;INIDICATE NO GST CNT TO INCREMENT
                00000000'GF   16  06AE  1211 65$:     JSB     G^MMG$CREDEL            ;USE THE COMMON CREATE PAGE CODE
                      04 50   E8  06B4  1212          BLBS    R0,66$
                10 AE   50    D0  06B7  1213          MOVL    R0,16(SP)               ;SAVE THE BAD STATUS
                00000000'GF   16  06BB  1214 66$:     JSB     G^MMG$RETRANGE
             59   52   51    C3  06C1  1215          SUBL3   R1,R2,R9                ;GET # BYTES ACTUALLY MAPPED
                        03    18  06C5  1216          BGEQ    67$                     ;BR IF RANGE IS IN INCREASING ORDER
                   59   59    CE  06C7  1217          MNEGL   R9,R9                   ;NEGATE THE BYTE COUNT, RANGE DECR ORDER
          59       59   F7 8F  78  06CA  1218 67$:     ASHL    #-9,R9,R9               ;CONVERT BYTE COUNT TO PAGE COUNT
                   0498 8F    BA  06CF  1219          POPR    #^M<R3,R4,R7,R10>       ;GET VA RANGE,SHB,PFNBASCNT,PFNBAS ADR
                        59    D6  06D3  1220          INCL    R9                      ;ACTUAL COUNT OF PAGES MAPPED
                      03 50   E8  06D5  1221          BLBS    R0,80$                  ;IF SUCCESSFUL USE SUCCESS STATUS
                              06D8  1222                                              ;ALREADY ON TOP OF STACK
                   6E   50    D0  06D8  1223 70$:     MOVL    R0,(SP)                 ;OTHERWISE SAVE ERROR STATUS
                              06DB  1224 ;
                              06DB  1225 ; 0(SP) = SYSTEM STATUS CODE
                              06DB  1226 ; R11 = GLOBAL SECTION DESCRIPTOR ADDRESS
                              06DB  1227 ;
                28   0A AB    91  06DB  1228 80$:     CMPB    GSD$B_TYPE(R11),#DYN$C_EXTGSD ;IS THIS AN EXTENDED GSD?
                        22    1F  06DF  1229          BLSSU   85$                     ;NO, BR TO CLEAN UP SECTION TBL ENTRY
                        59    1A  06E1  1230          BGTRU   PIECE_MAPPED            ;NO, BR IF SHARED MEMORY GSD
             50 00000000'GF   DE  06E3  1231 82$:     MOVAL   G^EXE$GL_GSDMTX,R0      ;GET MUTEX TO LOCK
             54 00000000'GF   D0  06EA  1232          MOVL    G^SCH$GL_CURPCB,R4      ;GET ADR OF PROCESS PCB
                00000000'GF   16  06F1  1233          JSB     G^SCH$LOCKW             ;LOCK GSD MUTEX FOR WRITING
                      2C AB   D7  06F7  1234          DECL    GSD$L_REFCNT(R11)       ;RELEASE THE GSD TO ALLOW DELETION
                       F903'  30  06FA  1235          BSBW    MMG$GSDMTXULK           ;UNLOCK GSD MUTEX
                        1F    11  06FD  1236          BRB     90$                     ;ALL DONE
                        59    D4  06FF  1237 84$:     CLRL    R9                      ;INDICATE NO PAGES MAPPED
                        D8    11  0701  1238          BRB     80$                     ;JOIN COMMON CODE
                51   16 AB    32  0703  1239 85$:     CVTWL   GSD$W_GSTX(R11),R1      ;SECTION INDEX
             55 00000000'FF   DE  0707  1240          MOVAL   @L^MMG$GL_SYSPHD,R5     ;SYSTEM PROCESS HEADER
                00000000'GF   16  070E  1241          JSB     G^MMG$DECSECREF         ;REMOVE SECTION REFERENCE BIAS
             54 00000000'GF   D0  0714  1242 88$:     MOVL    G^SCH$GL_CURPCB,R4      ;GET ADR OF PROCESS PCB
                       F8E2'  30  071B  1243          BSBW    MMG$DELGBLWCB           ;DELETE ANY GLOBAL WINDOWS
                        01    BA  071E  1244 90$:     POPR    #^M<R0>                 ;SYSTEM STATUS CODE
                              04  0720  1245          RET                             ;RETURN STATUS TO CALLER
                              0721  1246 ;
```

```
                                0721  1247 ; THE USER REQUESTED THAT NO PAGES BE MAPPED OR THERE WAS AN ERROR BEFORE
                                0721  1248 ; THE PAGES WERE MAPPED.  SHARED MEMORY GSD'S MUST BE MAPPED FOR CREATION; THIS
                                0721  1249 ; IS CHECKED IN $CRMPSC (FOR A VALID RELPAG, ONLY).
                                0721  1250 ;
                                0721  1251 PRE_MAP_ERR:
    6E    50      D0            0721  1252          MOVL    R0,(SP)                         ;SAVE RETURN ERROR CODE
28    0A AB      91            0724  1253 110$:     CMPB    GSD$B_TYPE(R11),#DYN$C_EXTGSD ;WHAT TYPE OF GSD IS THIS?
          D9      1F            0728  1254          BLSSU   85$                             ;BR IF LOCAL MEMORY
          B7      13            072A  1255          BEQL    82$                             ;BR IF EXTENDED GSD
    50    01      9A            072C  1256          MOVZBL  #1,R0                           ;ONE REF COUNT FOR A LOCK
00000000'GF       16            072F  1257          JSB     G^MMG$DECSHMREF                 ;RELEASE THE GSD LOCK
DB 66     00      E0            0735  1258          BBS     #GSD$V_VALID,GSD$L_GSDFL(R6),88$ ;BR IF NOT CREATING GS
        00D0      31            0739  1259          BRW     SHM_UNMAPPED                    ;BR TO DELETE GS AS CANNOT ALWAYS
                                073C  1260                                                  ;INITIALIZE A SHARED MEMORY GS.
                                073C  1261 ;
                                073C  1262 ; A PIECE OF A SHARED MEMORY GLOBAL SECTION HAS BEEN MAPPED.  UPDATE THE
                                073C  1263 ; RETURN RANGE OF VIRTUAL ADDRESSES MAPPED AND SET UP TO MAP THE NEXT PIECE.
                                073C  1264 ;
                                073C  1265 PIECE_MAPPED:
    50    59      D0            073C  1266          MOVL    R9,R0                           ;GET COUNT OF PTE'S JUST CREATED
          55      13            073F  1267          BEQL    140$                            ;BR IF NOTHING MAPPED
    56    5B      D0            0741  1268          MOVL    R11,R6                          ;RESTORE GSD ADR
OF 66     03      E0            0744  1269          BBS     #GSD$V_INITFAIL,GSD$L_GSDFL(R6),120$ ;BR IF UN-MAPPING GS
                                0748  1270                                                  ;R4=SHB ADDRESS
00000000'GF       16            0748  1271          JSB     G^MMG$INCSHMREF                 ;INCREMENT THE PROCESSOR REF COUNT
        04 AE     D5            074E  1272          TSTL    4(SP)                           ;IS START RETADR A VALID ADR?
          04      12            0751  1273          BNEQ    120$                            ;BR ON YES, FIRST ADR MAPPED SET
    04 AE   51    D0            0753  1274          MOVL    R1,4(SP)                        ;ASSUME RETADR NOT SPEC, USE INADR VALUE
          0C      BB            0757  1275 120$:     PUSHR   #^M<R2,R3>                      ;REMEM NEW START VA AND END VA
          6E      D6            0759  1276          INCL    (SP)                            ;ROUND NEW START VA TO PAGE BOUNDRY
          57      D7            075B  1277          DECL    R7                              ;ONE LESS PIECE OF SECTION TO MAP
          25      15            075D  1278          BLEQ    NO_MORE_PAGES                   ;BR IF NO MORE PIECES IN SECTION
              04  AA            D5    075F  1279          TSTL    4(R10)                          ;WAS SECTION CREATED IN PIECES?
              0D  13            0762  1280          BEQL    121$                            ;BR ON NO, ONE CONTIGUOUS BLOCK OF PAGES
7E    53  000001FF  8F          C9    0764  1281          BISL3   #^X1FF,R3,-(SP)          ;GET ENDING ADR, ROUNDED TO LAST BYTE
            8E      52          D1    076C  1282          CMPL    R2,(SP)+                ;IS SECTION MAPPED BACKWARDS?
              0D    1A          076F  1283          BGTRU   125$                            ;BR IF YES, RETURN ERROR CODE
        04 66       03          E0    0771  1284 121$:     BBS     #GSD$V_INITFAIL,GSD$L_GSDFL(R6),122$ ;BR IF UN-MAPPING GS
          0B 08 AE  E9          0775  1285          BLBC    8(SP),NO_MORE_PAGES             ;DON'T TRY TO MAP MORE IF GOT ERROR
              FDC2  31          0779  1286 122$:     BRW     MAP_NXT_BASE                    ;BR ON ANOTHER PIECE TO MAP/UNMAP
                                077C  1287
                                077C  1288
                    96          11    077C  1289 124$:     BRB     88$                     ;HOP, SKIP FOR BROKEN BRANCH DISPLACEMENT
                                077E  1290
                                077E  1291
    08 AE   0174    8F          3C    077E  1292 125$:     MOVZWL  #SS$_IVSSRQ,8(SP)       ;*** BACKWARDS MAPPING IS BROKEN
                                0784  1293                                                  ;*** FOR MULTIPLE PIECE SECTIONS
                                0784  1294 ;
                                0784  1295 ; NO MORE PIECES OF SHARED MEMORY SECTION LEFT TO MAP.  THIS MAY OR MAY NOT
                                0784  1296 ; BE AN ERROR, DEPENDING UPON WHETHER OR NOT AT LEAST ONE PAGE WAS MAPPED.
                                0784  1297 ;
                                0784  1298 NO_MORE_PAGES:
                    0C          BA    0784  1299          POPR    #^M<R2,R3>              ;CLEAN OFF START VA AND END VA
          50    F4  AD          D0    0786  1300          MOVL    MMG$L_SAVRETADR(FP),R0  ;GET ADDRESS OF RETURN ADR BUFFER
                    0A          13    078A  1301          BEQL    140$                    ;BR ON RETURN ADDRESS NOT REQUESTED
                                078C  1302          IFNOWRT #8,(R0),140$            ;CHECK IF BUFFER IS ACCESSIBLE
          60    04 AE           D0    0792  1303          MOVL    4(SP),(R0)              ;RESET TO VERY 1ST STARTVA IN RET RANGE
```

```
                50   01    9A  0796  1304 140$:   MOVZBL  #1,R0                        ;ONE REF COUNT FOR THE GSD LOCK
                                0799  1305                                            ;R4=SHB ADDRESS
00000000'GF     50   16      E4  0799  1306        JSB     G^MMG$DECSHMREF             ;REMOVE THE GSD LOCK
          69 66      03      E4  079F  1307        BBSC    #GSD$V_INITFAIL,GSD$L_GSDFL(R6),SHM_UNMAPPED ;BR IF INIT FAILED
          D5 66      00      E6  07A3  1308        BBSSI   #GSD$V_VALID,GSD$L_GSDFL(R6),124$ ;BR ON $MGBLSC REQUEST
                                07A7  1309 ;
                                07A7  1310 ; THE GLOBAL SECTION DESCRIPTOR WAS NOT VALID.  THEREFORE, THE GSD IS BEING
                                07A7  1311 ; MAPPED DUE TO A $CRMPSC REQUEST.  THE PAGES MUST NOW BE INITIALIZED.
                                07A7  1312 ;
                                07A7  1313                                            ;R6=GSD,INADR(AP),(SP)=MAP STATUS CODE
          52   04 AE      D0  07A7  1314        MOVL    4(SP),R2                    ;STARTVA FOR VERY FIRST PAGE MAPPED
                F852'      30  07AB  1315        BSBW    MMG$READ_GSD                ;BR TO READ SECTION INTO MEMORY
               41 50       E9  07AE  1316        BLBC    R0,RELEAS_SHMGS             ;BR IF FAILED TO INIT SECTION PAGES
                                07B1  1317 ;
                                07B1  1318 ; NOW THE SHARED MEMORY GSD TABLE MUST BE SEARCHED, CHECKING FOR SUCCESSFUL
                                07B1  1319 ; CREATION OF A GLOBAL SECTION WITH THE SAME NAME AS THE ONE BEING CREATED.
                                07B1  1320 ; THIS ALLOWS THE GSD TABLE TO BE OPEN FOR ACCESS DURING THE LENGTHY PROCESS
                                07B1  1321 ; OF INITIALIZING THE SECTION PAGES.  HOWEVER, IT ALSO ALLOWS TWO USERS TO
                                07B1  1322 ; TRY TO CREATE THE SAME SECTION AT THE SAME TIME.
                                07B1  1323 ;
                                07B1  1324                                            ;R11=GSD,R4=SHB
                 F84C'      30  07B1  1325        BSBW    MMG$UNIQUEGSD               ;VALIDATE THAT THE NEW GSD IS UNIQUE
                                07B4  1326                                            ;R5=SHD,R6=DUPLICATE GSD ADR
                   56       D5  07B4  1327        TSTL    R6                          ;IS THERE A DUPLICATE GSD?
                   29       13  07B6  1328        BEQL    135$                        ;BR IF THERE IS NOT A DUPLICATE
                                07B8  1329 ;
                                07B8  1330 ; NOW CHECK THAT THE DUPLICATE GLOBAL SECTION PASSES THE IDENT MATCH CONTROL
                                07B8  1331 ; TEST.  IF IT DOES NOT, THEN RETURN AN ERROR CODE TO THE USER.  IF IT DOES
                                07B8  1332 ; MATCH, THEN GO MAP THE DUPLICATE.
                                07B8  1333 ;
                18 A6       D5  07B8  1334        TSTL    GSD$L_IDENT(R6)             ;IS THIS ALWAYS MATCH?
                   27       13  07BB  1335        BEQL    141$                        ;BR TO USE SECTION IF ALWAYS MATCH
          1B AB    1B A6    91  07BD  1336        CMPB    GSD$L_IDENT+3(R6),GSD$L_IDENT+3(R11) ;DOES MAJOR ID MATCH?
                   0D       12  07C2  1337        BNEQ    133$                        ;BR ON NO, CAN'T USE DUP SECTION
          18 AB    18 A6    D1  07C4  1338        CMPL    GSD$L_IDENT(R6),GSD$L_IDENT(R11) ;COMPARE ENTIRE LONGWORD
                   06       1A  07C9  1339        BGTRU   133$                        ;BR IF LARGER, NO MATCH, CAN'T USE DUP
                15 18 A6    E9  07CB  1340        BLBC    GSD$L_IDENT(R6),141$        ;BR IF EXACT MATCH NOT REQUIRED
                   13       13  07CF  1341        BEQL    141$                        ;BR IF AN EXACT MATCH, CAN USE DUP
          6E    03F4 8F     3C  07D1  1342 133$:   MOVZWL  #SS$_IDMISMATCH,(SP)       ;REPORT DUPLICATE GS NAME
                50   01     9A  07D6  1343        MOVZBL  #1,R0                       ;ONE REFCNT FOR LOCK
           00000000'GF     16  07D9  1344        JSB     G^MMG$DECSHMREF             ;RELEASE LOCK ON DUP AS WON'T MAP TO IT
                   0A       11  07DF  1345        BRB     143$                        ;RETURN TO USER WITHOUT MAPPING DUP
                 0089       31  07E1  1346 135$:   BRW     150$                        ;BR ASSIST FOR NO DUP GSD PATH
             00 6B    04     E6  07E4  1347 141$:   BBSSI   #GSD$V_DUPGSD,GSD$L_GSDFL(R11),142$ ;RECORD DUP GSD CREATED
                6E    50     D0  07E8  1348 142$:   MOVL    R0,(SP)                     ;REMEMBER ERROR CODE
             04 AB    56     D0  07EB  1349 143$:   MOVL    R6,GSD$L_GSDBL(R11)         ;REMEMBER ADR OF DUP GSD TO MAP TO
                56    5B     D0  07EF  1350        MOVL    R11,R6                      ;RESTORE ADR OF GSD BEING CREATED
                                07F2  1351 ;
                                07F2  1352 ; AT THIS POINT, THE GLOBAL SECTION MUST BE RELEASED.  EITHER THE GLOBAL
                                07F2  1353 ; SECTION COULD NOT BE INITIALIZED OR A DUPLICATE SECTION WAS FOUND.  IF A
                                07F2  1354 ; DUPLICATE SECTION WAS FOUND, THE NEW SECTION MUST BE DELETED AND THE OLD
                                07F2  1355 ; SECTION MAPPED, INSTEAD.  THEREFORE, BEFORE RETURNING A STATUS CODE TO THE
                                07F2  1356 ; USER, THE PAGES MUST BE UNMAPPED FROM HIS VIRTUAL  ADDRESS SPACE.  THIS IS
                                07F2  1357 ; DONE BY SETTING A TEMP DELETE FLAG, AND THEN RE-EXECUTING THE $MGBLSC LOGIC
                                07F2  1358 ; CALLING $DELPAG INSTEAD OF $CREPAG.  WHEN THE LAST REFERENCE (PTE) TO THE
                                07F2  1359 ; SECTION IS DELETED, THEN THE SHARED MEMORY PAGES ASSOCIATED WITH THE SECTION
                                07F2  1360 ; AND THE GSD MAY BE RELEASED.  IF THERE IS A DUPLICATE SECTION, THEN A $MGBLSC
```

```
                        07F2  1361 ; IS PERFORMED TO IT.
                        07F2  1362 ;
                        07F2  1363 RELEAS_SHMGS:
            04 AE   D5  07F2  1364          TSTL    4(SP)                           ;WAS ANYTHING MAPPED?
              15    13  07F5  1365          BEQL    SHM_UNMAPPED                    ;BR ON NO, NOTHING TO UNMAP
         00 66  03    E3  07F7  1366          BBCS    #GSD$V_INITFAIL,GSD$L_GSDFL(R6),144$ ;SET TMP DEL INDIC
           50   01    9A  07FB  1367 144$:    MOVZBL  #1,R0                           ;REF COUNT INCREMENT
                        07FE  1368                                                  ;R4=SHB ADDRESS
       00000000'GF      16  07FE  1369          JSB     G^MMG$INCSHMREF                 ;LOCK GSD SO IT CAN'T BE DELETED
              53    DD  0804  1370 145$:    PUSHL   R3                              ;LAST ENDING VA MAPPED
            08 AE   DD  0806  1371          PUSHL   8(SP)                           ;FIRST STARTING VA MAPPED
             FD23   31  0809  1372          BRW     MAPGBLSEC4                      ;REPEAT CODE, UNMAPPING THE VA SPACE
                        080C  1373 SHM_UNMAPPED:
    55  00000000'EF   D0  080C  1374          MOVL    MMG$GL_SYSPHD,R5                ;GET ADR OF SYSTEM PROCESS HEADER
      51    16 A6    32  0813  1375          CVTWL   GSD$W_GSTX(R6),R1               ;GET SECTION TABLE ENTRY INDEX
    50  55    20 A5   C1  0817  1376          ADDL3   PHD$L_PSTBASOFF(R5),R5,R0 ;COMPUTE ADR OF SECTION TBL ENTRY
       5A   6041    DE  081C  1377          MOVAL   (R0)[R1],R10                    ;SO THAT IT CAN BE RELEASED
              6A    D4  0820  1378          CLRL    SEC$L_GSD(R10)                  ;INDICATE NO GSD CONNECTED TO IT
                        0822  1379          ASSUME  SEC$L_PAGCNT EQ SEC$L_REFCNT+4
            18 AA   7C  0822  1380          CLRQ    SEC$L_REFCNT(R10)               ;CLEAR REFCNT, SO IT WILL BE DELETED
    00 14 AA    0E    E5  0825  1381          BBCC    #SEC$V_PERM,SEC$W_FLAGS(R10),151$ ;CLEAR THE PERMANENT FLAG
              01    E6  082A  1382 151$:    BBSSI   #PHD$V_DALCSTX,-                ;LET SOMEONE IN PROPER
         00 36 A5       082C  1383                  PHD$W_FLAGS(R5),152$            ;STATE DELETE IT (HOLDING GSD MUTEX)
           50   01    9A  082F  1384 152$:    MOVZBL  #SHD$V_BITMAPLCK,R0             ;NUMBER OF BIT LOCK REQUESTED
             F7CB'  30  0832  1385          BSBW    MMG$SHMTXLK                     ;REQUEST MUTEX AND BIT LOCK
                        0835  1386 ; **********
                        0835  1387 ; AT SOME TIME THIS SHOULD SEND A MESSAGE TO THE ERROR LOGGER.
                        0835  1388 ; **********
            10 50   E9  0835  1389          BLBC    R0,146$                         ;UNABLE TO ACQUIRE BIT MAP LOCK
       55   04 A4   D0  0838  1390          MOVL    SHB$L_DATAPAGE(R4),R5           ;GET ADR OF COMMON DATA PAGE
             F7C1'  30  083C  1391          BSBW    MMG$SET_BITMAP                  ;RELEASE ALL PAGES ALLOCATED FOR GS
             F7BE'  30  083F  1392          BSBW    MMG$SHMTXULK                    ;RELEASE SHM MUTEX
00 009F C5    01    E7  0842  1393          BBCCI   #SHD$V_BITMAPLCK,SHD$B_FLAGS(R5),146$ ;RELEASE BIT LCK
      00 66   00    E7  0848  1394 146$:    BBCCI   #GSD$V_VALID,GSD$L_GSDFL(R6),147$ ;SET THE GSD NOT VALID
              5B    D4  084C  1395 147$:    CLRL    R11                             ;ASSUME NO DUPLICATE GSD TO MAP TO
      04 66   04    E7  084E  1396          BBCCI   #GSD$V_DUPGSD,GSD$L_GSDFL(R6),148$ ;BR IF NO DUP GSD TO MAP TO
      5B   04 A6    D0  0852  1397          MOVL    GSD$L_GSDBL(R6),R11             ;GET ADR OF DUPLICATE GSD
      00 66   01    E7  0856  1398 148$:    BBCCI   #GSD$V_LOCKED,GSD$L_GSDFL(R6),149$ ;UNLOCK THE GSD FOR REUSE
      51   15 A4   9A  085A  1399 149$:    MOVZBL  SHB$B_PORT(R4),R1               ;GET PORT # FOR THIS PROCESSOR
   3C A541    01    58  085E  1400          ADAWI   #1,SHD$W_GSDQUOTA(R5)[R1] ;GIVE BACK SHM GSD QUOTA FOR PORT
            0C A4   D7  0863  1401          DECL    SHB$L_REFCNT(R4)                ;ONE LESS SHM GSD OWNED BY THIS PORT
         56   5B    D0  0866  1402          MOVL    R11,R6                          ;SET ADR OF DUPLICATE GSD TO MAP TO
              06    13  0869  1403          BEQL    160$                            ;BR TO RETURN ERROR CODE
              97    11  086B  1404          BRB     145$                            ;BR TO MAP TO DUPLICATE GSD
      00 6B   01    E7  086D  1405 150$:    BBCCI   #GSD$V_LOCKED,GSD$L_GSDFL(R11),160$ ;UNLOCK THE GSD FOR USE
             FEA0   31  0871  1406 160$:    BRW     88$                             ;ALL DONE, GO CLEAN UP
                        0874  1407          .DSABL  LSB
```

```
                        0874  1409          .SBTTL FAST_MAP - DO COMMON CASES EFFICIENTLY
                        0874  1410 ;
                        0874  1411 ; MAP A GLOBAL SECTION THE EASY WAY
                        0874  1412 ;
                        0874  1413 ;       This fast path is used for new adress space beyond the end of a
                        0874  1414 ;       region with all quota checks made in advance.
                        0874  1415 ;       The main loop does four pages at a time to further decrease the
                        0874  1416 ;       loop overhead.
                        0874  1417 ;
                        0874  1418 FAST_MAP:
            7E  52  7D  0874  1419          MOVQ    R2,-(SP)                ;SAVE THE VA'S
            53  52  D1  0877  1420          CMPL    R2,R3                   ;IS R2 THE LOWEST VA
                03  1F  087A  1421          BLSSU   10$                     ;YES
            52  53  D0  087C  1422          MOVL    R3,R2                   ;NOW R2 IS LOWEST
                        087F  1423
         07 52  1E  E1  087F  1424 10$:     BBC     #VA$V_P1,R2,20$         ;BRANCH IF P0 SPACE
                        0883  1425 ;
                        0883  1426 ; P1 SPACE
                        0883  1427 ;
        53  00D0 C5  DE  0883  1428          MOVAL   PHD$L_P1BR(R5),R3       ;ADR OF POINTER TO P1PT
                05  11  0888  1429          BRB     30$
                        088A  1430 ;
                        088A  1431 ; P0 SPACE
                        088A  1432 ;
        53  00C8 C5  DE  088A  1433 20$:    MOVAL   PHD$L_P0BR(R5),R3       ;ADR OF POINTER TO P0PT
                        088F  1434
          18 AA  56  C0  088F  1435 30$:    ADDL    R6,SEC$L_REFCNT(R10)    ;ADJUST REFERENCE COUNT
                        0893  1436
     1A 14 AA  0D  E1  0893  1437          BBC     #SEC$V_RESIDENT,SEC$W_FLAGS(R10),35$ ;IS IT RESIDENT
 59   58  16  00  EF  0898  1438          EXTZV   #PTE$V_GPTX,#PTE$S_GPTX,R8,R9 ;EXTRACT GPTX FROM PROPOSED PTE
59  00000000'FF49  DE  089D  1439          MOVAL   @MMG$GL_GPTBASE[R9],R9  ;GET FIRST GPTE ADDRESS
                59  D6  08A5  1440          INCL    R9                      ;INDICATOR FOR FIRST TIME THROUGH LOOP
 58   867FFFFF  8F  CA  08A7  1441          BICL    #^C<PTE$M_OWN!PTE$M_PROT>,R8 ;LEAVE OWNER-PROT IN PTE
       00 58  1F  E2  08AE  1442          BBSS    #PTE$V_VALID,R8,35$      ;VALID
                        08B2  1443
 51   52  15  09  EF  08B2  1444 35$:     EXTZV   #VA$V_VPN,#VA$S_VPN,R2,R1 ;VIRTUAL PAGE NUMBER
 50   56  02  00  EF  08B7  1445          EXTZV   #0,#2,R6,R0             ;PAGES TO LEAVE A MULTIPLE OF 4
                29  13  08BC  1446          BEQL    50$
                        08BE  1447
     0C 14 AA  0D  E0  08BE  1448          BBS     #SEC$V_RESIDENT,SEC$W_FLAGS(R10),45$ ;IS IT RESIDENT
      00 B341  88  9E  08C3  1449 40$:     MOVAB   (R8)+,@(R3)[R1]         ;STORE NEW PTE
                51  D6  08C8  1450          INCL    R1                      ;NEXT PAGE
            F6  50  F5  08CA  1451          SOBGTR  R0,40$
                18  11  08CD  1452          BRB     50$                     ;REJOIN COMMON CODE
                        08CF  1453
      00000011'EF  16  08CF  1454 45$:     JSB     LCKPAGTBL1
 58   15  00  69  F0  08D5  1455          INSV    (R9),#PTE$V_PFN,#PTE$S_PFN,R8 ;PUT PFN IN PTE
      00 B341  58  D0  08DA  1456          MOVL    R8,@(R3)[R1]            ;STORE NEW PTE
            59  04  C0  08DF  1457          ADDL    #4,R9                   ;NEXT GPTE
                51  D6  08E2  1458          INCL    R1                      ;NEXT PAGE
            E8  50  F5  08E4  1459          SOBGTR  R0,45$
                        08E7  1460
  56   56  FE  8F  78  08E7  1461 50$:     ASHL    #-2,R6,R6
                03  12  08EC  1462          BNEQ    55$
              007D  31  08EE  1463          BRW     70$                     ;DONE
                        08F1  1464
     21 14 AA  0D  E0  08F1  1465 55$:     BBS     #SEC$V_RESIDENT,SEC$W_FLAGS(R10),65$ ;IS IT RESIDENT
```

```
                00 B341    88    9E  08F6  1466 60$:    MOVAB   (R8)+,@(R3)[R1]          ;STORE NEW PTE
                           51    D6  08FB  1467         INCL    R1                       ;NEXT PAGE
                00 B341    88    9E  08FD  1468         MOVAB   (R8)+,@(R3)[R1]          ;STORE NEW PTE
                           51    D6  0902  1469         INCL    R1                       ;NEXT PAGE
                00 B341    88    9E  0904  1470         MOVAB   (R8)+,@(R3)[R1]          ;STORE NEW PTE
                           51    D6  0909  1471         INCL    R1                       ;NEXT PAGE
                00 B341    88    9E  090B  1472         MOVAB   (R8)+,@(R3)[R1]          ;STORE NEW PTE
                           51    D6  0910  1473         INCL    R1                       ;NEXT PAGE
                      E1   56    F5  0912  1474         SOBGTR  R6,60$
                           57    11  0915  1475         BRB     70$                      ;REJOIN COMMON CODE
                                     0917  1476
                 00000011'EF    16  0917  1477 65$:    JSB     LCKPAGTBL1
             58  15    00       69  F0  091D  1478         INSV    (R9),#PTE$V_PFN,#PTE$S_PFN,R8 ;PUT PFN IN PTE
                00 B341    58    D0  0922  1479         MOVL    R8,@(R3)[R1]             ;STORE NEW PTE
                      59   04    C0  0927  1480         ADDL    #4,R9                    ;NEXT GPTE
                           51    D6  092A  1481         INCL    R1                       ;NEXT PAGE
                                     092C  1482 ;
                 00000011'EF    16  092C  1483         JSB     LCKPAGTBL1               ;LOCK THE PAGE TABLE
             58  15    00       69  F0  0932  1484         INSV    (R9),#PTE$V_PFN,#PTE$S_PFN,R8 ;PUT PFN IN PTE
                00 B341    58    D0  0937  1485         MOVL    R8,@(R3)[R1]             ;STORE NEW PTE
                      59   04    C0  093C  1486         ADDL    #4,R9                    ;NEXT GPTE
                           51    D6  093F  1487         INCL    R1                       ;NEXT PAGE
                                     0941  1488 ;
                 00000011'EF    16  0941  1489         JSB     LCKPAGTBL1               ;LOCK THE PAGE TABLE
             58  15    00       69  F0  0947  1490         INSV    (R9),#PTE$V_PFN,#PTE$S_PFN,R8 ;PUT PFN IN PTE
                00 B341    58    D0  094C  1491         MOVL    R8,@(R3)[R1]             ;STORE NEW PTE
                      59   04    C0  0951  1492         ADDL    #4,R9                    ;NEXT GPTE
                           51    D6  0954  1493         INCL    R1                       ;NEXT PAGE
                                     0956  1494 ;
                 00000011'EF    16  0956  1495         JSB     LCKPAGTBL1               ;LOCK THE PAGE TABLE
             58  15    00       69  F0  095C  1496         INSV    (R9),#PTE$V_PFN,#PTE$S_PFN,R8 ;PUT PFN IN PTE
                00 B341    58    D0  0961  1497         MOVL    R8,@(R3)[R1]             ;STORE NEW PTE
                      59   04    C0  0966  1498         ADDL    #4,R9                    ;NEXT GPTE
                           51    D6  0969  1499         INCL    R1                       ;NEXT PAGE
                                     096B  1500 ;
                         A9 56    F5  096B  1501         SOBGTR  R6,65$
                                     096E  1502
                      51   8E    7D  096E  1503 70$:    MOVQ    (SP)+,R1                 ;GET BACK THE VA'S
                      52   51    D1  0971  1504         CMPL    R1,R2
                           12    1A  0974  1505         BGTRU   90$                      ;GOING BACKWARDS
                           04    1F  0976  1506         BLSSU   80$                      ;GOING FORWARDS
                   0C  52   1E    E0  0978  1507         BBS     #VA$V_P1,R2,90$          ;EQUAL - P1 IS BACKWARDS
                51 01FF 8F       AA  097C  1508 80$:    BICW    #^X1FF,R1                ;STARTVA IS START OF PAGE
                52 01FF 8F       A8  0981  1509         BISW    #^X1FF,R2                ;ENDVA IS END OF PAGE
                           0A    11  0986  1510         BRB     100$
                52 01FF 8F       AA  0988  1511 90$:    BICW    #^X1FF,R2                ;STARTVA IS START OF PAGE
                51 01FF 8F       A8  098D  1512         BISW    #^X1FF,R1                ;ENDVA IS END OF PAGE
                      50   01    3C  0992  1513 100$:   MOVZWL  #SS$_NORMAL,R0           ;INDICATE SUCCESSFUL COMPLETION
                           05    05  0995  1514         RSB
```

```
                     0996  1516          .SBTTL  SETSECPROTOWN - SET SECTION PROTECTION AND OWNER
                     0996  1517 ;
                     0996  1518 ; INPUTS:
                     0996  1519 ;
                     0996  1520 ;       R6  = GLOBAL SECTION DESCRIPTOR ADDRESS
                     0996  1521 ;       R8  = PAGE TABLE ENTRY WITH SECTION INDEX, FLAGS, PAGE TYPE BITS
                     0996  1522 ;             OR PAGE TABLE ENTRY WITH PFN FOR PFNMAP-ING
                     0996  1523 ;       R9  = SECTION FLAGS
                     0996  1524 ;       R10 = SECTION TABLE ENTRY ADDRESS OR 0 IF GSD TYPE IS EXTENDED GSD
                     0996  1525 ;       MMG$L_MAXACMODE(FP) = MAXIMIZED ACCESS MODE
                     0996  1526 ;
                     0996  1527 ; OUTPUTS:
                     0996  1528 ;
                     0996  1529 ;       R0 = SYSTEM STATUS CODE
                     0996  1530 ;       R8 OWNER AND PROTECTION FIELDS FILLED IN
                     0996  1531 ;
                     0996  1532 SETSECPROTOWN:
                     0996  1533          .ENABL  LSB
              01     DD    0996  1534          PUSHL   #SS$_NORMAL                     ;ASSUME NORMAL SUCCESS RETURN CODE
     51 59  02  06   EF    0998  1535          EXTZV   #SEC$V_WRTMOD,#SEC$S_WRTMOD,R9,R1 ;GET WRTMOD MAPPER SPECIFIED
              5A     D5    099D  1536          TSTL    R10                             ;IS THERE A SECTION TABLE ENTRY?
              08     13    099F  1537          BEQL    2$                              ;BR IF NO SECTION TABLE
           02 06     EF    09A1  1538          EXTZV   #SEC$V_WRTMOD,#SEC$S_WRTMOD,- ;GET ACCESS MODE FROM SECTION
        50   14 AA         09A4  1539                  SEC$W_FLAGS(R10),R0             ;TABLE ENTRY, I.E., WHAT CREATOR SPEC
              0A     11    09A7  1540          BRB     4$                              ;JOIN COMMON CODE
              56     D5    09A9  1541 2$:      TSTL    R6                              ;IS THERE A GLOBAL SECTION DESCRIPTOR?
              1E     13    09AB  1542          BEQL    10$                             ;BR ON LOCAL PFNMAP, CREATOR=MAPPER
           02 06     EF    09AD  1543          EXTZV   #SEC$V_WRTMOD,#SEC$S_WRTMOD,- ;GET ACCESS MODE FROM GSD
        50   20 A6         09B0  1544                  GSD$W_FLAGS(R6),R0              ;I.E., WHAT CREATOR OF GS SPECIFIED
           51   50    D1    09B3  1545 4$:      CMPL    R0,R1                           ;NOW MINIMIZE THE WRITE ACCESS
              08     15    09B6  1546          BLEQ    5$                              ;BR IF MAPPER SPECIFIED MORE ACCESS
        6E    203C 8F     3C    09B8  1547          MOVZWL  #SS$_IVLVEC,(SP)                ;SET ALTERNATE RETURN SUCCESS CODE
           50   51    D0    09BD  1548          MOVL    R1,R0                           ;USE CREATOR'S ACCESS
51    FC AD    02   00    EF    09C0  1549 5$:      EXTZV   #0,#2,B^MMG$L_MAXACMODE(FP),R1 ;GET READ ACCESS MODE
           51   50    D1    09C6  1550          CMPL    R0,R1                           ;IF WRTMOD > ACMOD, THEN USE ACMOD
              03     19    09C9  1551          BLSS    15$                             ;TO PREVENT ILLEGAL ACCESS, E.G., ERUW
           50   51    D0    09CB  1552 10$:     MOVL    R1,R0                           ;AND TO SET CORRECT OWNER FIELD
     58    02   17    50    F0    09CE  1553 15$:     INSV    R0,#PTE$V_OWN,#PTE$S_OWN,R8 ;SET PAGE OWNER IN PTE
                     09D3  1554 ;
                     09D3  1555 ; CALCULATE PROTECTION FIELD
                     09D3  1556 ;
        26 59  03    E1    09D3  1557          BBC     #SEC$V_WRT,R9,NO_WRT_ACCESS ;BR IF NOT TRYING TO WRITE SECTION
              5A     D5    09D7  1558          TSTL    R10                             ;IS THERE A SECTION TABLE ENTRY?
              0B     12    09D9  1559          BNEQ    20$                             ;YES, GO CHECK ITS FLAGS
              56     D5    09DB  1560          TSTL    R6                              ;IS THIS A GLOBAL SECTION?
              0C     13    09DD  1561          BEQL    30$                             ;NO, THEN MUST BE LOCAL PFNMAP
                           09DF  1562                                                  ;WITH NO GSD OR SECTION TBL ENTRY (ONLY
                           09DF  1563                                                  ;FLAGS ARE IN R9 AND ALREADY TESTED)
     2A 20 A6    03    E1    09DF  1564          BBC     #SEC$V_WRT,GSD$W_FLAGS(R6),100$ ;BRANCH IF ILLEGAL TO WRITE
              05     11    09E4  1565          BRB     30$                             ;GO GET ACCESS MODE
     23 14 AA    03    E1    09E6  1566 20$:     BBC     #SEC$V_WRT,SEC$W_FLAGS(R10),100$ ;BRANCH IF ILLEGAL TO WRITE
           50   03    C4    09EB  1567 30$:     MULL2   #3,R0                           ;SCALE WRITE ACCESS MODE BY THREE BITS
50    FB'AF   03    50    EE    09EE  1568          EXTV    R0,#3,B^WRTMOD_TBL,R0       ;GET PTE CODE FOR THIS WRITE MODE
              0A     18    09F4  1569          BGEQ    GET_READ_ACCESS                 ;BR IF NOT USER MODE WRITE
           50   04    9A    09F6  1570          MOVZBL  #PRT$C_UW,R0                    ;SET USER MODE WRITE, SPECIAL CASE
              0A     11    09F9  1571          BRB     SET_PTE_PROT                    ;GO SET PROTECTION FIELD OF PTE
                           09FB  1572 WRTMOD_TBL:
```

```
                        080A  09FB          .WORD   ^B100000001010          ;3 BIT PTE CODES, INDEXED BY 3 * WRTMOD
                              09FD  NO_WRT_ACCESS:
                50   03   9A  09FD          MOVZBL  #3,R0                   ;SET PTE CODE FOR READ ONLY ACCESS
                              0A00  GET_READ_ACCESS:
      50   02   02   51   F0  0A00          INSV    R1,#2,#2,R0             ;INSERT READ ACCESS BESIDE WRITE ACCESS
                              0A05  SET_PTE_PROT:
      58   04   1B   50   F0  0A05          INSV    R0,#PTE$V_PROT,#PTE$S_PROT,R8 ;PUT PROTECTION IN PTE
                50   8E   D0  0A0A  90$:    MOVL    (SP)+,R0                ;SET RETURN STATUS CODE
                         05  0A0D          RSB
                              0A0E  ;
                              0A0E  ; TRIED TO MAP READ ONLY SECTION WRITABLE
                              0A0E  ;
                6E   24   3C  0A0E  100$:   MOVZWL  #SS$_NOPRIV,(SP)        ;SET RETURN ERROR CODE, NO PRIVILEGE
                     F7   11  0A11          BRB     90$                     ;RETURN ERROR CODE
                              0A13          .DSABL  LSB
```

```
                                0A13          .SBTTL  INITSECTBL - ALLOC & INIT SECTION TABLE ENTRY
                                0A13  ;
                                0A13  ; INPUTS:
                                0A13  ;
                                0A13  ;       R5 = PROCESS HEADER ADDRESS
                                0A13  ;       R6 = GLOBAL SECTION DESCRIPTOR ADDRESS IF GLOBAL SECTION
                                0A13  ;          = CHANNEL CONTROL BLOCK ADDRESS IF PROCESS SECTION
                                0A13  ;       R7 = NUMBER OF PAGES TO BE MAPPED
                                0A13  ;       R8 = CHANNEL CONTROL BLOCK ADDRESS
                                0A13  ;       R9 = SECTION FLAGS
                                0A13  ;
                                0A13  ; OUTPUTS:
                                0A13  ;
                                0A13  ;       R0  = SYSTEM STATUS CODE
                                0A13  ;       R1  = SECTION TABLE INDEX
                                0A13  ;       R2,R3 ALTERED
                                0A13  ;       R7  = SECTION PAGE COUNT
                                0A13  ;       R10 = SECTION TABLE ADDRESS
                                0A13  ;
                                0A13  ;
                                0A13  ; ***** NOTE THAT SECTION SIZE MUST BE SUCH THAT SECTION INDICES HAVE THE
                                0A13  ; ***** LOW BIT ZERO.
                                0A13  ;
                                0A13          ASSUME  SEC$C_LENGTH@-2&1 EQ 0
                                0A13
                                0A13          .ENABL  LSB
                                0A13  ;
                                0A13  ; GLOBAL SECTION BEING CREATED ON CHANNEL WITH PROCESS SECTION INDEX
                                0A13  ; GET WINDOW ADDRESS
                                0A13  ;
50   00000000'9F    D0          0A13  10$:    MOVL    @#CTL$GL_PHD,R0          ;PROCESS HEADER ADDRESS
     50    20 A0    C0          0A1A          ADDL    PHD$L_PSTBASOFF(R0),R0   ;FORM BASE OF PROCESS SECTION TABLE
     52    0C A042  D0          0A1E          MOVL    SEC$L_WINDOW(R0)[R2],R2  ;GET WINDOW ADDRESS FROM PROCESS SECTION
              7E    11          0A23          BRB     100$                     ;AND REJOIN THE NORMAL FLOW
                                0A25  ;
                                0A25  ; SPECIAL CASE CODE FOR NON-FCP WINDOW - I.E. ONE CREATED WITHOUT BENEFIT OF
                                0A25  ; THE ACP - USUALLY DONE AT SYSTEM INITIALIZATION TIME. THE LOOP IS TO
                                0A25  ; TAKE CARE OF POSSIBLY NON-CONTIGUOUS FILES.
                                0A25  ;
              51    DD          0A25  40$:    PUSHL   R1                       ;NEED A SCRATCH REGISTER
              7E    D4          0A27          CLRL    -(SP)                    ;INITILIZE TOTAL VBN'S
     51    16 A2    3C          0A29          MOVZWL  WCB$W_NMAP(R2),R1        ;PICK UP NUMBER OF MAPPING POINTERS
     52    30 A2    9E          0A2D          MOVAB   WCB$W_P1_COUNT(R2),R2    ;POINT AT FIRST ONE
        50    62    3C          0A31  45$:    MOVZWL  (R2),R0                  ;PICK UP THE NUMBER OF BLOCKS/POINTER
        6E    50    C0          0A34          ADDL2   R0,(SP)                  ;ADD TO TOTAL MAPPED
        52    06    C0          0A37          ADDL2   #6,R2                    ;ADVANCE TO NEXT POINTER
          F4 51     F5          0A3A          SOBGTR  R1,45$                   ;TAKE CARE OF THEM ALL
        50    8E    7D          0A3D          MOVQ    (SP)+,R0                 ;RESTORE R1.  R0 = TOTAL VBNS IN FILE
           0091     31          0A40          BRW     120$                     ;AND REJOIN THE MAIN PATH CODE
                                0A43  ;
                                0A43  ; CHANNEL IS ACTIVE OR OTHERWISE INAPPROPRIATE FOR CREATING A SECTION
                                0A43  ;
     7E    01CC 8F  3C          0A43  50$:    MOVZWL  #SS$_NOTFILEDEV,-(SP)    ;FILE NOT RND, FOD, OR DIR
              05    11          0A48          BRB     70$                      ;GO RELEASE SECTION TABLE ENTRY
     7E    026C 8F  3C          0A4A  60$:    MOVZWL  #SS$_IVCHNLSEC,-(SP)     ;INVALID CHANNEL FOR SECTION
     00000000'GF    16          0A4F  70$:    JSB     G^MMG$DALCSTX            ;DEALLOCATE SECTION TABLE ENTRY
              01    BA          0A55          POPR    #^M<R0>                  ;GET ERROR STATUS
```

```
                                05  0A57  1646 80$:     RSB
                                    0A58  1647
                          00FB  31  0A58  1648 81$:     BRW     230$
                            B6  11  0A5B  1649 82$:     BRB     10$
                                    0A5D  1650
                                    0A5D  1651 INITSECTBL:
                 G0000000'GF  16  0A5D  1652            JSB     G^MMG$ALCSTX                    ;ALLOCATE SECTION TABLE INDEX
                        F1 50  E9  0A63  1653            BLBC    R0,80$                          ;BRANCH IF NONE AVAILABLE
                53   55   20 A5  C1  0A66  1654            ADDL3   PHD$L_PSTBASOFF(R5),R5,R3 ;BASE ADDRESS OF SECTION TABLE
                       5A   6341  DE  0A6B  1655            MOVAL   (R3)[R1],R10            ;ADDRESS OF SECTION TABLE ENTRY
                          6A   56  D0  0A6F  1656            MOVL    R6,(R10)                ;CCB ADDRESS OR GSD ADDRESS
                        14 AA   59  B0  0A72  1657            MOVW    R9,SEC$W_FLAGS(R10)     ;SET FLAGS
                               58  D5  0A76  1658            TSTL    R8
                               DE  13  0A78  1659            BEQL    81$                     ;NO CHANNEL - SKIP ALL THIS STUFF
                            0A A8  B5  0A7A  1660            TSTW    CCB$W_IOC(R8)           ;IF ANY OUTSTANDING I/O ON CHANNEL
                               CB  12  0A7D  1661            BNEQ    60$                     ;THEN IT CANNOT BE USED
                          50   68  D0  0A7F  1662            MOVL    CCB$L_UCB(R8),R0        ;GET UCB FOR DEVICE CHARACTERISTICS
50   10004008 8F    38 A0  CB  0A82  1663            BICL3   UCB$L_DEVCHAR(R0), -    ;CHECK THAT DEVICE HAS DIRECTORIES,
                                    0A8B  1664                    #<DEV$M_DIR!DEV$M_FOD!DEV$M_RND>,R0 ;FILES, AND IS RANDOM-ACCESS
                               B6  12  0A8B  1665            BNEQ    50$                     ;BRANCH IF ANY CHARACTERISTIC IS MISSING
                        52   04 A8  D0  0A8D  1666            MOVL    CCB$L_WIND(R8),R2       ;WINDOW ADDRESS FROM CHANNEL
                               B7  13  0A91  1667            BEQL    60$                     ;BRANCH IF NO FILE IS OPEN
                            B4 52  E8  0A93  1668            BLBS    R2,60$                  ;CAN'T USE CHAN IF ACCESS/DEACCESS PENDING
                               0B  19  0A96  1669            BLSS    100$                    ;BRANCH IF WINDOW ADDRESS
                          52   52  32  0A98  1670            CVTWL   R2,R2                   ;FORM PROCESS SECTION INDEX
                             BD 59  E8  0A9B  1671            BLBS    R9,82$                  ;BRANCH IF CREATING A GLOBAL SECTION
                    52   0C A342  D0  0A9E  1672            MOVL    SEC$L_WINDOW(R3)[R2],R2 ;FETCH WINDOW FROM PROCESS SECTION
                      0C AA   52  D0  0AA3  1673 100$:     MOVL    R2,SEC$L_WINDOW(R10)    ;SET WINDOW ADDRESS IN SECTION TABLE
                  0F  0B A2   01  E0  0AA7  1674            BBS     #WCB$V_WRITE,WCB$B_ACCESS(R2),105$ ;BR IF FILE WRITE ACCESSED
                     0F 59    03  E1  0AAC  1675            BBC     #SEC$V_WRT,R9,110$      ;BRANCH IF NOT MAPPING FOR WRITE
                     0B 59    01  E0  0AB0  1676            BBS     #SEC$V_CRF,R9,110$      ;WRITE AND CRF IS OK FOR READ ONLY FILE
                   7E   03FC 8F  3C  0AB4  1677            MOVZWL  #SS$_NOWRT,-(SP)        ;ERROR, CANNOT CREATE WRITABLE SECTION
                               94  11  0AB9  1678            BRB     70$                     ;TO A READ ONLY FILE, RETURN TO USER
                                    0ABB  1679
                                    0ABB  1680            ASSUME  SEC$V_WRT LE 7
                        14 AA   08  88  0ABB  1681 105$:     BISB    #SEC$M_WRT,SEC$W_FLAGS(R10) ;NOTE SECTION WRITABLE
                        50   18 A2  D0  0ABF  1682 110$:     MOVL    WCB$L_FCB(R2),R0        ;FCB ADDRESS FROM WINDOW
                               03  12  0AC3  1683            BNEQ    112$                    ;GOT A REAL ONE
                             FF5D  31  0AC5  1684            BRW     40$                     ;BRANCH IF NONE THERE
                  03 14 A2     0B  E2  0AC8  1685 112$:     BBSS    #WCB$V_NOTRUNC,WCB$W_ACON(R2),115$ ;DISALLOW TRUNCATE ON THE FILE
                        20 A0   B6  0ACD  1686            INCW    FCB$W_TCNT(R0)          ;COUNT TRUNCATE LOCKS
                        50   E4 AD  D0  0AD0  1687 115$:     MOVL    B^MMG$L_EFBLK(FP),R0    ;LAST VIRTUAL BLOCK THAT MAY BE MAPPED
52   28 AC     01  C3  0AD4  1688 120$:     SUBL3   #1,VBN(AP),R2           ;DESIRED STARTING VBN - 1
                               02  18  0AD9  1689            BGEQ    130$                    ;IF WAS SPECIFIED AS 0
                               52  D4  0ADB  1690            CLRL    R2                      ;THEN MAKE IT VBN 1
10 AA    52     01  C1  0ADD  1691 130$:     ADDL3   #1,R2,SEC$L_VBN(R10)    ;SET STARTING VBN IN SECTION TABLE
                               57  D5  0AE2  1692            TSTL    R7                      ;NUMBER OF PAGES IN SECTION
                                    0AE4  1693                                            ;STARTING AT SPECIFIED VBN
                               03  14  0AE4  1694            BGTR    140$                    ;BRANCH IF NOT DEFAULTED TO "ENTIRE FILE"
                        57     50  D0  0AE6  1695            MOVL    R0,R7                   ;USE ENTIRE FILE
                   7E   52     57  C1  0AE9  1696 140$:     ADDL3   R7,R2,-(SP)             ;FORM HIGHEST VBN TO BE MAPPED
                        8E     50  D1  0AED  1697            CMPL    R0,(SP)+                ;TRYING TO MAP BEYOND EOF?
                               06  14  0AF0  1698            BGTR    150$                    ;BRANCH IF NOT
                   57   50     52  C3  0AF2  1699            SUBL3   R2,R0,R7                ;FORM NEW PAGCNT = MAXVBN - (STARTVBN - 1)
                               6B  15  0AF6  1700            BLEQ    290$                    ;BRANCH IF NOTHING TO MAP, EOF
                     1C AA     57  D0  0AF8  1701 150$:     MOVL    R7,SEC$L_PAGCNT(R10)    ;SET SECTION SIZE
                           11 59  E9  0AFC  1702            BLBC    R9,170$                 ;BRANCH IF PROCESS SECTION
```

```
                     0AFF  1703 ;
                     0AFF  1704 ; GLOBAL SECTION - ONCE THE WINDOW IS MADE SHARED, DALCSTX CANNOT BE
                     0AFF  1705 ; USED TO DELETE THE SECTION TABLE ENTRY.  DALCSTXSCN MUST BE USED
                     0AFF  1706 ; BECAUSE IT RELEASES THE WINDOW.
                     0AFF  1707 ;
   50    0C AA   D0  0AFF  1708          MOVL    SEC$L_WINDOW(R10),R0    ;WINDOW ADDRESS
03 0B A0    03   E2  0B03  1709          BBSS    #WCB$V_SHRWCB,WCB$B_ACCESS(R0),160$ ;MAKE INTO SHARED WINDOW
                     0B08  1710                                          ;BRANCH IF ALREADY A SHARED WINDOW
         0354    30  0B08  1711          BSBW    MMG$RET_BYT_QUOTA       ;RESTORE BYTCNT QUOTA TO FILE OWNER
      0E A0      B6  0B0B  1712 160$:    INCW    WCB$W_REFCNT(R0)        ;ANOTHER REFERENCE FOR SECTION ENTRY
         0A      11  0B0E  1713          BRB     180$
                     0B10  1714 ;
                     0B10  1715 ; PROCESS SECTION
                     0B10  1716 ;
   50    04 A8   D0  0B10  1717 170$:    MOVL    CCB$L_WIND(R8),R0       ;GET WINDOW OR SECTION FROM CHANNEL
         09      14  0B14  1718          BGTR    190$                    ;BRANCH IF SECTION INDEX
   04 A8    51   3C  0B16  1719          MOVZWL  R1,CCB$L_WIND(R8)       ;1ST PROCESS SECTION ON CHANNEL
                     0B1A  1720                                          ;STORE SECTION INDEX IN CCB
      50    5A   D0  0B1A  1721 180$:    MOVL    R10,R0                  ;SECTION TABLE ENTRY ADDRESS
         10      11  0B1D  1722          BRB     200$                    ;INIT FORWARD AND BACKWARD SECTION INDICES
                     0B1F  1723 ;
                     0B1F  1724 ; SUBSEQUENT SECTION (NOT THE FIRST) ON THIS CHANNEL
                     0B1F  1725 ;
      50    50   32  0B1F  1726 190$:    CVTWL   R0,R0                   ;INSERT AFTER THIS SECTION
   06 AA    50   B0  0B22  1727          MOVW    R0,SEC$W_SECXBL(R10)    ;CURSEC(BL) = BAKSECX
    50    6340   DE  0B26  1728          MOVAL   (R3)[R0],R0             ;ADR OF BACKWARD SECTION TABLE ENTRY
04 AA    04 A0   B0  0B2A  1729          MOVW    SEC$W_SECXFL(R0),SEC$W_SECXFL(R10) ;CURSEC(FL) = BAKSEC(FL)
   04 A0    51   B0  0B2F  1730 200$:    MOVW    R1,SEC$W_SECXFL(R0)     ;BAKSEC(FL) = CURSECX
   50    04 AA   32  0B33  1731          CVTWL   SEC$W_SECXFL(R10),R0    ;FORWARD SECTION INDEX (FORSECX)
    50    6340   DE  0B37  1732          MOVAL   (R3)[R0],R0             ;FORWARD SECTION TABLE ENTRY ADDRESS
   06 A0    51   B0  0B3B  1733          MOVW    R1,SEC$W_SECXBL(R0)     ;FORSEC(BL) = CURSECX
   18 AA    01   D0  0B3F  1734          MOVL    #1,SEC$L_REFCNT(R10)    ;NO REFERENCES YET
   50    30 AC   98  0B43  1735          CVTBL   PFC(AP),R0              ;GET PAGE FAULT CLUSTER
         04      18  0B47  1736          BGEQ    210$                    ;BRANCH IF NOT TOO BIG
   50    7F 8F   9A  0B49  1737          MOVZBL  #127,R0                 ;IT WAS TOO BIG, SET TO MAX
                     0B4D  1738
                     0B4D  1739          ASSUME  SEC$B_PFC EQ SEC$L_VPXPFC+3
08 AA    50    18   78  0B4D  1740 210$:    ASHL    #24,R0,SEC$L_VPXPFC(R10) ;SET PFC, ZERO VIRTUAL PAGE INDEX
         50    01   3C  0B52  1741          MOVZWL  #SS$_NORMAL,R0          ;SUCCESSFUL COMPLETION
               05   05  0B55  1742          RSB
                     0B56  1743 ;
                     0B56  1744 ; PAGE FILE BACKING STORE
                     0B56  1745 ;
   14 AA    08   88  0B56  1746 230$:    BISB    #SEC$M_WRT,SEC$W_FLAGS(R10) ;NOTE SECTION WRITABLE
   1C AA    57   D0  0B5A  1747          MOVL    R7,SEC$L_PAGCNT(R10)    ;SET SECTION SIZE
      0C AA      D4  0B5E  1748          CLRL    SEC$L_WINDOW(R10)       ;NO WINDOW - PAGE FILE BACKING STORE
         B7      11  0B61  1749          BRB     180$
                     0B63  1750 ;
                     0B63  1751 ; ATTEMPT TO CREATE A SECTION BEYOND END OF FILE
                     0B63  1752 ;
7E   0870 8F   3C  0B63  1753 290$:    MOVZWL  #SS$_ENDOFFILE,-(SP)    ;SET RETURN STATUS
      FEE4      31  0B68  1754          BRW     70$                     ;CLEAN UP AND EXIT
                     0B6B  1755
                     0B6B  1756          .DSABL  LSB
```

```
                                0B6B  1758          .SBTTL  MAP PROCESS SECTION
                                0B6B  1759 ;
                                0B6B  1760 ; CALLING SEQUENCE:
                                0B6B  1761 ;
                                0B6B  1762 ;       BRW     MAP_PROCESS_SECTION
                                0B6B  1763 ;
                                0B6B  1764 ; INPUTS:
                                0B6B  1765 ;
                                0B6B  1766 ;       R4 = PCB ADDRESS
                                0B6B  1767 ;       R8 = CHANNEL CONTROL BLOCK ADDRESS, IF PFNMAP FLAG IS CLEAR
                                0B6B  1768 ;       R9 = SECTION FLAGS
                                0B6B  1769 ;       0(SP) = STARTVA
                                0B6B  1770 ;       4(SP) = ENDVA
                                0B6B  1771 ;       8(SP) = SUCCESS CODE FOR MAP SECTION
                                0B6B  1772 ;       PFN(AP) = FIRST PFN TO MAP TO, IF PFNMAP FLAG IS SET
                                0B6B  1773 ;
                                0B6B  1774 MAP_PROCESS_SEC:
                                0B6B  1775          SETIPL  #IPL$_ASTDEL            ;NO AST'S WHILE MANIPULATING HEADER
                          5B    D4  0B6E  1776          CLRL    R11                     ;INDICATE NO GSD ADR
                55  00000000'9F   D0  0B70  1777          MOVL    @#CTL$GL_PHD,R5         ;PROCESS HEADER ADDRESS
                  0A 59     10    E1  0B77  1778          BBC     #SEC$V_PFNMAP,R9,1$     ;IS PFNMAPPING REQUESTED?
                  06 65     1A    E0  0B7B  1779          BBS     #PRV$V_PFNMAP,PHD$Q_PRIVMSK(R5),1$ ;BR ON HAVE PRIV
                     50     24    3C  0B7F  1780          MOVZWL  #SS$_NOPRIV,R0          ;NO PRIVILEGE FOR REQUESTED OPERATION
                        0139   31    82  1781          BRW     90$                     ;RETURN ERROR CODE TO CALLER
                00000000'GF   16    85  1782 1$:       JSB     G^MMG$DALCSTXSCN        ;SCAN FOR SECTIONS TO DEALLOCATE
                     56   58    D0  0B8B  1783          MOVL    R8,R6                   ;CCB ADDRESS
                                0B8E  1784
                  57   24 AC    D0  0B8E  1785          MOVL    SECPAGCNT(AP),R7        ;GET # OF PAGES IN SECTION
                          59    DD  0B92  1786          PUSHL   R9                      ;REMEMBER FLAGS
                                0B94  1787          ASSUME  <SEC$M_CRF!SEC$M_DZRO> EQ 6
          03    59   02    01    ED  0B94  1788          CMPZV   #1,#2,R9,#<<SEC$M_CRF!SEC$M_DZRO>@-1> ;IF CRF+DZRO, THEN DONT
                          24    13  0B99  1789          BEQL    8$                      ;CREATE SECTION TABLE ENTRY JUST PTES
                   24 59    10    E0  0B9B  1790          BBS     #SEC$V_PFNMAP,R9,10$    ;DON'T CREATE SECTION PTE FOR PFNMAP-ING
                                0B9F  1791          ASSUME  <<SEC$M_DZRO!SEC$M_CRF!SEC$M_WRT>&^XFFFFFF00> EQ 0
                      59    59    9A  0B9F  1792          MOVZBL  R9,R9                   ;ELIM GBL FLAGS, PFNMAP, & EXPREG
                         FEB8   30  0BA2  1793          BSBW    INITSECTBL              ;ALLOCATE AND INIT A SECTION TABLE ENTRY
                        03 50   E8  0BA5  1794          BLBS    R0,5$                   ;BR IF SECTION TABLE ENTRY CREATED
                         0113   31  0BA8  1795          BRW     90$                     ;BR IF NONE AVAILABLE OR ERROR
                                0BAB  1796 ;
                                0BAB  1797 ; R1  = SECTION TABLE INDEX
                                0BAB  1798 ; R7  = SECTION PAGE COUNT
                                0BAB  1799 ; R10 = SECTION TABLE ENTRY ADDRESS
                                0BAB  1800 ;
                                0BAB  1801
                                0BAB  1802          ASSUME  SEC$V_WRT EQ SEC$V_DZRO+1
                                0BAB  1803          ASSUME  SEC$V_DZRO EQ SEC$V_CRF+1
       58    14 AA    03    01    EF  0BAB  1804 5$:       EXTZV   #SEC$V_CRF,#3,SEC$W_FLAGS(R10),R8 ;GET CRF, DZRO, WRT BITS
                58    0440 8F    A8  0BB1  1805          BISW    #<PTE$M_TYP1 ! PTE$M_TYP0>@-16,R8 ;OR IN TYPE BITS
                58    58   10    9C  0BB6  1806          ROTL    #16,R8,R8               ;PUT IN HIGH 16 BITS
                58    58   51    B0  0BBA  1807          MOVW    R1,R8                   ;SET SECTION INDEX
                           13    11  0BBD  1808          BRB     15$                     ;SKIP PFNMAP-ING PTE CREATION
                           58    D4  0BBF  1809 8$:       CLRL    R8                      ;USE DEMAND-ZERO PTE FORMAT
                           0B    11  0BC1  1810          BRB     12$                     ;CONTINUE MAPPING SECTION
                   58   28 AC    D0  0BC3  1811 10$:      MOVL    PFN(AP),R8              ;GET STARTING PFN FOR SECTION
             58    80200000 8F    C8  0BC7  1812          BISL    #^X<PTE$M_VALID ! PTE$M_WINDOW>,R8 ;SET VALID AND WINDOW BITS
                           5A    D4  0BCE  1813 12$:      CLRL    R10                     ;INDICATE NO SECTION TABLE ENTRY
                           56    D4  0BD0  1814          CLRL    R6                      ;NO GSD ADDRESS
```

```
                FDC1   30  0BD2  1815 15$:    BSBW    SETSECPROTOWN                ;SET SECTION PROTECTION AND OWNER
              06 50    E8  0BD5  1816         BLBS    R0,15$                       ;BRANCH IF GOOD PROTECTION CODE
              F4 AD    D4  0BD8  1817         CLRL    MMG$L_SAVRETADR(FP)          ;DON'T RETURN AN ADDRESS RANGE
                009D   31  0BDB  1818         BRW     20$                          ; THE MOVQ (SP)+ WILL TAKE GARBAGE FROM STAC
           50     01    D1  0BDE  1819 15$:    CMPL    #SS$_NORMAL,R0               ;WAS ALTERNATE SUCCESS CODE RETURNED?
                  04    13  0BE1  1820         BEQL    16$                          ;BR IF NORMAL CODE RETURNED
        0C AE     50    D0  0BE3  1821         MOVL    R0,12(SP)                    ;SAVE ALTERNATE CODE TO RETURN TO CALLER
                  50  8ED0  0BE7  1822 16$:    POPL    R0                           ;GET FLAGS BACK
                  0C    BA  0BEA  1823         POPR    #^M<R2,R3>                   ;R2=STARTVA, R3=ENDVA
        28 50     11    E1  0BEC  1824         BBC     #SEC$V_EXPREG,R0,17$         ;BR IF RANGE IS EXPLICITLY STATED
                            0BF0  1825 ;
                            0BF0  1826 ; FIND THE FIRST AVAILABLE VIRTUAL ADDRESS AND COMPUTE THE RANGE TO BE MAPPED.
                            0BF0  1827 ;
     51   57    09    78  0BF0  1828         ASHL    #9,R7,R1                     ;CONVERT PAGE COUNT TO # OF
                  51    D7  0BF4  1829         DECL    R1                           ;BYTES BETWEEN START VA AND END VA
  50    00000000'9F    D0  0BF6  1830         MOVL    @#CTL$GL_PHD,R0              ;GET PROCESS HEADER FOR PROCESS
     0F 52      1E    E1  0BFD  1831         BBC     #VA$V_P1,R2,17$              ;BR IF MAPPING INTO P0 SPACE
53   30 A0   000001FF  8F  C1  0C01  1832         ADDL3   #^X1FF,PHD$L_FREP1VA(R0),R3 ;ENDING VA IN P1 SPACE
     52   53    51    C3  0C0A  1833         SUBL3   R1,R3,R2                     ;STARTING VA IN P1 SPACE
                  08    11  0C0E  1834         BRB     17$                          ;JOIN COMMON CODE
     52   28 A0    D0  0C10  1835 17$:    MOVL    PHD$L_FREP0VA(R0),R2         ;STARTING VA IN P0 SPACE
     53   52    51    C1  0C14  1836         ADDL3   R1,R2,R3                     ;ENDING VA IN P0 SPACE
                  5A    D5  0C18  1837 17$:    TSTL    R10                          ;IS IT PFN MAPPING
                  39    13  0C1A  1838         BEQL    19$                          ;YES - DO IT THE HARD WAY
           54     52    7D  0C1C  1839         MOVQ    R2,R4                        ;GET THE START AND END VA
        00000000'EF    16  0C1F  1840         JSB     MMG$IN_REGION                ;IS IT NEW ADDRESS SPACE
              2D 50    E9  0C25  1841         BLBC    R0,19$                       ;NO - CREATE IT THE HARD WAY
           57   56    D1  0C28  1842         CMPL    R6,R7                        ;CHECK FIT OF SECTION IN THE SPACE
                  28    12  0C2B  1843         BNEQ    19$                          ;NOT EXACT - DO IT THE HARD WAY
     54    00000000'EF    D0  0C2D  1844         MOVL    SCH$GL_CURPCB,R4
      5A     20 A5    C2  0C34  1845         SUBL    PHD$L_PSTBASOFF(R5),R10      ;NORMALIZE SECTION ADDRESS IN CASE IT MOVES
        00000000'EF    16  0C38  1846         JSB     MMG$TRY_ALL                  ;SEE IF REGION CAN BE EXPANDED
              10 50    E9  0C3E  1847         BLBC    R0,18$                       ;NO - CREATE IT THE HARD WAY
        00000000'EF    16  0C41  1848         JSB     MMG$FAST_CREATE              ;DO IT THE FAST WAY
      5A     20 A5    C0  0C47  1849         ADDL    PHD$L_PSTBASOFF(R5),R10      ;UN-NORMALIZE SECTION ADDRESS
       18 AA   57    C0  0C4B  1850         ADDL    R7,SEC$L_REFCNT(R10)         ;ACCOUNT FOR THE PAGES MAPPED
                  24    11  0C4F  1851         BRB     19$
      5A     20 A5    C0  0C51  1852 18$:    ADDL    PHD$L_PSTBASOFF(R5),R10      ;UN-NORMALIZE SECTION ADDRESS
     59   57    01    C3  0C55  1853 19$:    SUBL3   #1,R7,R9                     ;SECTION PAGE COUNT BASE 0
           030C 8F    BB  0C59  1854 19$:    PUSHR   #^M<R2,R3,R8,R9>             ;NEEDED TO REDO OPERATION
     56   0CE1'CF    DE  0C5D  1855         MOVAL   W^MMG$MAPSECPAG,R6           ;MAP SECTION PAGE ROUTINE
                  0A    E2  0C62  1856         BBSS    #MMG$V_NOWAIT_IPL0,-         ;RETURN INSTEAD OF WAITING AT IPL 0
           00 FC AD          0C64  1857                 MMG$L_MAXACMODE(FP),19$
        00000000'GF    16  0C67  1858 19$:    JSB     G^MMG$CREDEL                 ;CREATE THE PAGES
           2C   50    B1  0C6D  1859         CMPW    R0,#SS$_ABORT                ;DID WE ABORT RATHER THAN WAIT AT IPL 0
                  4D    13  0C70  1860         BEQL    100$                         ;YES - GO CLEAN UP
              5E   10    C0  0C72  1861         ADDL    #<4*4>,SP                    ;REMOVE SAVED REGISTERS
              7E   51    7D  0C75  1862 19$:    MOVQ    R1,-(SP)                     ;SAVE RETURN ADDRESS RANGE
                 04 50    E8  0C78  1863         BLBS    R0,30$                       ;IF SUCCESSFUL, USE SUCCESS CODE AT 8(SP)
        08 AE     50    D0  0C7B  1864 20$:    MOVL    R0,8(SP)                     ;OTHERWISE SAVE ERROR CODE
                  5A    D5  0C7F  1865 30$:    TSTL    R10                          ;IS THERE A SECTION TABLE ENTRY?
                  2D    13  0C81  1866         BEQL    85$                          ;BR ON NO SECTION TABLE ENTRY
1C AA    18 AA    01    C3  0C83  1867         SUBL3   #1,SEC$L_REFCNT(R10),SEC$L_PAGCNT(R10) ;SET ACTUAL PAGE COUNT
                  1C    13  0C89  1868         BEQL    80$                          ;BRANCH IF NO PAGES MAPPED
                            0C8B  1869 ;
                            0C8B  1870 ; AT LEAST ONE PAGE WAS MAPPED.
                            0C8B  1871 ; R1, R2 CONTAIN THE RETURN RANGE VALUES
```

```
                                0C8B  1872 ;
                  03 51    E9   0C8B  1873          BLBC    R1,40$                  ;BRANCH IF R1 IS THE LOWEST ADDRESS
                  51  52    D0   0C8E  1874          MOVL    R2,R1                   ;R2 WAS THE LOWEST
        50   51   16  09    EE   0C91  1875 40$:     EXTV    #VA$V_VPN,#VA$S_VPN+1,R1,R0 ;GET VIRTUAL PAGE INDEX
                      0B    18   0C96  1876          BGEQ    50$                     ;BRANCH IF NOT P1 SPACE
   51   00000000'EF   07    9C   0C98  1877          ROTL    #7,SGN$GL_PTPAGCNT,R1   ;OFFSET IN LONG WORD FROM BEGIN OF PAGE TABL
                  50  51    C0   0CA0  1878          ADDL    R1,R0                   ;CORRECT P1 SPACE INDEX
              08 AA   50    C8   0CA3  1879 50$:     BISL    R0,SEC$L_VPXPFC(R10)    ;SET VIRTUAL PAGE INDEX, PRESERVE PFC
                  51  58    32   0CA7  1880 80$:     CVTWL   R8,R1                   ;SECTION TABLE INDEX
            00000000'GF     16   0CAA  1881          JSB     G^MMG$DECSECREF         ;COUNT ONE LESS SECTION REFERENCE
                  51  8E    7D   0CB0  1882 85$:     MOVQ    (SP)+,R1                ;GET BACK RETURN RANGE
            00000000'GF     16   0CB3  1883          JSB     G^MMG$RETRANGE          ;RETURN ADDRESS RANGE TO CALLER
                  02 50     E9   0CB9  1884          BLBC    R0,90$                  ;ERROR - USE THIS STATUS
                      01    BA   0CBC  1885          POPR    #^M<R0>                 ;RETURN STATUS CODE
                            04   0CBE  1886 90$:     RET                             ;AND RETURN TO CALLER
                                0CBF  1887 ;
                                0CBF  1888 ; THERE WAS A PAGE WITH I/O IN PROGRESS WHICH WE MUST WAIT FOR.
                                0CBF  1889 ; DELETE THE PARTIALLY MAPPED SECTION FIRST
                                0CBF  1890 ;
                      0A    E5   0CBF  1891 100$:    BBCC    #MMG$V_NOWAIT_IPL0,-    ;WAIT THIS TIME
                00 FC AD        0CC1  1892                  MMG$L_MAXACMODE(FP),110$
                  52  51    7D   0CC4  1893 110$:    MOVQ    R1,R2                   ;COMPLETED ADDRESS RANGE
                  53  57    C0   0CC7  1894          ADDL    R7,R3                   ;ADD IN THE PROBLEM PAGE
        56  00000000'EF     DE   0CCA  1895          MOVAL   MMG$DELPAG,R6           ;DELETE ONLY THIS PASS
            00000000'EF     16   0CD1  1896          JSB     MMG$CREDEL
                  9B 50     E9   0CD7  1897          BLBC    R0,194$                 ;ERROR
                030C 8F     BA   0CDA  1898          POPR    #^M<R2,R3,R8,R9>
                   FF78     31   0CDE  1899          BRW     191$                    ;GO TRY AGAIN
```

```
OCE1  1901          .SBTTL  MAPSECPAG - MAP A SINGLE PROCESS/GLOBAL SECTION PAGE
OCE1  1902
OCE1  1903  ;++
OCE1  1904  ; FUNCTIONAL DESCRIPTION:
OCE1  1905  ;
OCE1  1906  ;       MAPSECPAG MAPS A SINGLE PAGE OF A GLOBAL OR PROCESS SECTION AT THE
OCE1  1907  ; SPECIFIED VITUAL ADDRESS.  AS IN CREPAG, THE PAGE TABLE IS EXTENDED
OCE1  1908  ; IF NECESSARY (WHICH COULD FAIL IF THE VIRTUAL ADDRESS SPACE IS FULL),
OCE1  1909  ; AND THE PAGE TABLE ENTRY IS DELETED IF NECESSARY AND THEN SET TO
OCE1  1910  ; SPECIFIED VALUE.
OCE1  1911  ;
OCE1  1912  ;       THE CREATE PAGE LOGIC WILL CAUSE THE SECTION TABLE TO BE RELOCATED
OCE1  1913  ; IF IT IS NECESSARY TO EXPAND THE PROCESS HEADER.  THIS HAPPENS WHEN THE
OCE1  1914  ; WORKING SET IS FULL AND IT IS BACK-TO-BACK WITH THE SECTION TABLE.  THAT
OCE1  1915  ; IS WHY THE SECTION TABLE ENTRY ADDRESS MUST BE RECOMPUTED AFTER EACH
OCE1  1916  ; CALL TO MMG$CREPAG.  THE SECTION TABLE INDEX NEVER CHANGES, BUT THE OFFSET
OCE1  1917  ; TO THE SECTION TABLE FROM THE TOP OF THE PROCESS HEADER DOES.
OCE1  1918  ;
OCE1  1919  ;       THIS ROUTINE IS ALSO USED TO UNMAP THE PAGES OF SHARED MEMORY GLOBAL
OCE1  1920  ; SECTION WHICH FOR SOME REASON COULD NOT BE INITIALIZED.  THE BIT,
OCE1  1921  ; GSD$V_INITFAIL, INDICATES THAT THE RANGE OF VIRTUAL ADDRESS SPACE IS TO
OCE1  1922  ; BE DELETED INSTEAD OF CREATED.  THIS SERVICE WILL ONLY BE NEEDED FOR SHARED
OCE1  1923  ; MEMORY GLOBAL SECTIONS SINCE ONLY THIS TYPE OF SECTION IS INITIALIZED BEFORE
OCE1  1924  ; THE $CRMPSC STATUS CODE CAN BE RETURNED TO THE USER.  THE NEW CONTENTS OF
OCE1  1925  ; THE PAGE TABLE ENTRY (R8) IS NOT USED WHEN DELETING VIRTUAL ADDRESS SPACE.
OCE1  1926  ;
OCE1  1927  ; CALLING SEQUENCE:
OCE1  1928  ;
OCE1  1929  ;       BSBW    MMG$MAPSECPAG
OCE1  1930  ;
OCE1  1931  ; INPUT PARAMETERS:
OCE1  1932  ;
OCE1  1933  ;       R0 = MODE FOR CREATING NEW PAGE
OCE1  1934  ;       R2 = VIRTUAL ADDRESS OF PAGE TO CREATE
OCE1  1935  ;       R4 = PCB ADDRESS
OCE1  1936  ;       R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
OCE1  1937  ;       R6 = COUNT-1 OF PAGES TO BE MAPPED ACCORDING TO THE INPUT RANGE
OCE1  1938  ;       R7 = +^X200 IF MAPPING FORWARDS IN THE VIRTUAL ADDRESS SPACE
OCE1  1939  ;          = -^X200 IF MAPPING BACKWARDS IN THE VIRTUAL ADDRESS SPACE
OCE1  1940  ;       R8 = NEW CONTENTS OF PAGE TABLE ENTRY
OCE1  1941  ;       R9 = COUNT-1 OF PAGES LEFT IN THE SECTION THAT COULD BE MAPPED
OCE1  1942  ;       R10 = SECTION TABLE ENTRY ADDRESS OR 0 IF NONE APPLIES
OCE1  1943  ;       R11 = GLOBAL SECTION DESCRIPTOR ADDRESS OR 0 IF NONE APPLIES
OCE1  1944  ;
OCE1  1945  ;       THE CURRENT IPL MUST BE AT ASTDEL
OCE1  1946  ;
OCE1  1947  ; IMPLICIT INPUTS:
OCE1  1948  ;
OCE1  1949  ;       NONE
OCE1  1950  ;
OCE1  1951  ; OUTPUT PARAMETERS:
OCE1  1952  ;
OCE1  1953  ;       R0 = ERROR STATUS CODE
OCE1  1954  ;       R2 PRESERVED
OCE1  1955  ;       R1,R3-R7,R9 DESTROYED
OCE1  1956  ;
OCE1  1957  ; IMPLICIT OUTPUTS:
```

```
                     OCE1  1958 ;
                     OCE1  1959 ;       PTE CORRESPONDING TO SPECIFIED VIRTUAL ADDRESS IS DELETED AND
                     OCE1  1960 ;       THE DESIRED PTE IS STORED
                     OCE1  1961 ;
                     OCE1  1962 ;       IF PAGE TABLE EXPANSION IS NECESSARY THEN THE FOLLOWING
                     OCE1  1963 ;       ARE AFFECTED:
                     OCE1  1964 ;
                     OCE1  1965 ;       PHD$L_FREPOVA OR PHD$L_FREP1VA  ;1ST FREE PAGE AT END OF P0/P1 PAGE TABLE
                     OCE1  1966 ;       PHD$L_P0LRASTL OR PHD$L_P1LR    ;LENGTH OF PT IN HARDWARE PCB
                     OCE1  1967 ;       PR$_P0LR OR PR$_P1LR            ;LENGTH OF PT IN PROCESSOR REG
                     OCE1  1968 ;       PHD$L_FREPTECNT                 ;FREE PTE COUNTER
                     OCE1  1969 ;
                     OCE1  1970 ; COMPLETION CODES:
                     OCE1  1971 ;
                     OCE1  1972 ;       SS$_NORMAL                      ;SUCCESSFUL COMPLETION
                     OCE1  1973 ;       SS$_NOPRIV                      ;NO PRIVILEGE TO CREATE/DELETE PAGE
                     OCE1  1974 ;       SS$_VASFULL                     ;VIRTUAL ADDRESS SPACE FULL
                     OCE1  1975 ;
                     OCE1  1976 ; SIDE EFFECTS:
                     OCE1  1977 ;       NONE
                     OCE1  1978 ;
                     OCE1  1979 ;--
                     OCE1  1980
                     OCE1  1981 ;
                     OCE1  1982 ; ****************************************************************************
                     OCE1  1983 ;
                     OCE1  1984 ; ***************** THE FOLLOWING CODE MAY BE PAGED *****************
                     OCE1  1985 ;
                00000CE1  1986       .PSECT  YF$$SYSCRMPSC
                     OCE1  1987 ;
                     OCE1  1988 ; ****************************************************************************
                     OCE1  1989 ;
                     OCE1  1990 MMG$MAPSECPAG:
          59   56   D1  OCE1  1991       CMPL    R6,R9                   ;MORE PAGES THAN IN THE SECTION?
               03   15  OCE4  1992       BLEQ    10$                     ;BRANCH IF NOT
          56   59   D0  OCE6  1993       MOVL    R9,R6                   ;YES, USE SECTION SIZE
59   57   F7   8F   78  OCE9  1994 10$:   ASHL    #-9,R7,R9               ;+1 OR -1 DEPENDING ON DIRECTION
               5B   D5  OCEE  1995       TSTL    R11                     ;IS THIS A GLOBAL SECTION MAPPING?
               53   12  OCF0  1996       BNEQ    50$                     ;YES, BRANCH IF IS A GLOBAL SECTION
                     OCF2  1997
                     OCF2  1998 ;
                     OCF2  1999 ; LOGIC FOR PROCESS SECTIONS ONLY:
                     OCF2  2000 ;
                     OCF2  2001 ;       PTE'S THAT MAP A PROCESS SECTION ARE FILLED IN WITH
                     OCF2  2002 ;       THE SECTION TABLE INDEX, AND THEREFORE, NO INCREMENT OR
                     OCF2  2003 ;       DECREMENT IS NEEDED FOR SEQUENTIAL PTE'S.
                     OCF2  2004 ;
               5A   D5  OCF2  2005       TSTL    R10                     ;CHECK IF NO SECTION TABLE ENTRY
               24   13  OCF4  2006       BEQL    30$                     ;BR ON NONE, PFNMAP SECTION OR DZERO
               59   D4  OCF6  2007       CLRL    R9                      ;NO INC FOR NORMAL PROCESS SEC PTE'S
                     OCF8  2008
                     OCF8  2009 ;
                     OCF8  2010 ; LOGIC FOR ALL SECTIONS, EXCEPT PFNMAP AND SHARED MEMORY:
                     OCF8  2011 ;
                     OCF8  2012 ;       THIS IS THE CREATE-PAGE LOOP FOR ALL SECTION MAPPING,
                     OCF8  2013 ;       EXCEPT FOR PFNMAP SECTIONS.
                     OCF8  2014 ;       RESIDENT GLOBAL SECTIONS ARE A SPECIAL CASE OF THIS
```

```
                                    0CF8  2015 ;
                60 14 AA    0D      E0  0CF8  2016 20$:     BBS     #SEC$V_RESIDENT,SEC$W_FLAGS(R10),60$ ;RESIDENT GLOBAL SECTION
                F0 AD    02'AF      DE  0CFD  2017          MOVAL   B^25$,B^MMG$L_PAGESUBR(FP) ;SKIP INIT CODE FOR SUBSEQUENT PAGES
                   5A    20 A5      C2  0D02  2018 25$:     SUBL    PHD$L_PSTBASOFF(R5),R10 ;SUBTRACT OFF POINTER THAT MIGHT CHANGE
                 00000000'GF      16  0D06  2019          JSB     G^MMG$CREPAG            ;CREATE AND STORE THE PAGE TABLE ENTRY
                   5A    20 A5      C0  0D0C  2020 26$:     ADDL    PHD$L_PSTBASOFF(R5),R10 ;ADD IN OFFSET THAT MAY HAVE CHANGED
                         06 50      E9  0D10  2021          BLBC    R0,29$                  ;BRANCH IF ERROR
                         18 AA      D6  0D13  2022          INCL    SEC$L_REFCNT(R10)       ;COUNT USES OF THIS SECTION
                      58    59      C0  0D16  2023          ADDL    R9,R8                   ;COMPUTE NEXT PTE'S CONTENTS
                                    05  0D19  2024 29$:     RSB
                                        0D1A  2025 ;
                                        0D1A  2026 ; LOGIC FOR PFNMAP, SHARED MEMORY SECTIONS, AND DZERO CRF
                                        0D1A  2027 ;
                                        0D1A  2028 ;       THIS CODE ASSUMES THAT THERE IS NO SECTION TABLE ENTRY.
                                        0D1A  2029 ;
                                        0D1A  2030 ;       FOR SHARED MEMORY, THERE IS A SECTION TABLE ENTRY BUT THE REFERENCE
                                        0D1A  2031 ;       COUNT IS NOT INCREMENTED FOR EACH PAGE MAPPING TO IT.  INSTEAD, THE
                                        0D1A  2032 ;       REFERENCE COUNT IS KEPT IN THE SHARED MEMORY GSD.  THEREFORE, R10 IS
                                        0D1A  2033 ;       0 FOR SHARED MEMORY MAPPING REQUESTS.
                                        0D1A  2034 ;
                      02 58    15      E0  0D1A  2035 30$:     BBS     #PTE$V_WINDOW,R8,40$    ;PFN SECTION OR DZERO CRF?
                               59      D4  0D1E  2036          CLRL    R9                      ;DZERO
                F0 AD    25'AF      DE  0D20  2037 40$:     MOVAL   B^45$,B^MMG$L_PAGESUBR(FP) ;SKIP INIT CODE FOR SUBSEQUENT PAGES
                 00000000'GF      16  0D25  2038 45$:     JSB     G^MMG$CREPAG            ;CREATE AND STORE THE PAGE TABLE ENTRY
                         16 50      E9  0D2B  2039          BLBC    R0,49$                  ;BRANCH IF ERROR
                      58    59      C0  0D2E  2040          ADDL    R9,R8                   ;COMPUTE NEXT PTE'S CONTENTS
                   0F 58    15      E1  0D31  2041          BBC     #PTE$V_WINDOW,R8,49$    ;BR IF NOT PFNMAP
00000000'EF       58    15   00      ED  0D35  2042          CMPZV   #PTE$V_PFN,#PTE$S_PFN,R8,MMG$GL_MAXMEM ;CHECK PFN
                               04      1B  0D3E  2043          BLEQU   49$                     ;MEMORY IS SEEN BY MP SECONDARY
                         0104 C5      D6  0D40  2044          INCL    PHD$L_MPINHIBIT(R5)     ;LOCK PROCESS ONTO PRIMARY PROCESSOR
                                    05  0D44  2045 49$:     RSB                             ;
                                        0D45  2046
                                        0D45  2047 ;
                                        0D45  2048 ; LOGIC FOR SHMGSD UNMAP ONLY:
                                        0D45  2049 ;
                                        0D45  2050 ;       THIS CODE IS USED TO UNMAP A SHARED MEMORY GLOBAL SECTION,
                                        0D45  2051 ;       WHEN THE RACE CONDITION OCCURS CAUSING TWO SECTIONS OF THE
                                        0D45  2052 ;       SAME NAME ARE CREATED IN ONE SHARED MEMORY.  THE SECOND
                                        0D45  2053 ;       SECTION CREATED, IS UNMAPPED AND RELEASED.
                                        0D45  2054 ;
                   28    0A AB      91  0D45  2055 50$:     CMPB    GSD$B_TYPE(R11),#DYN$C_EXTGSD ;IS THIS A PFNMAPPED SECTION?
                               D5      13  0D49  2056          BEQL    40$                     ;BR IF PFNMAP
                               AB      1F  0D4B  2057          BLSSU   20$                     ;BR IF NORMAL LOCAL MEMORY GSD
                      CF 6B    03      E1  0D4D  2058          BBC     #GSD$V_INITFAIL,GSD$L_GSDFL(R11),40$ ;BR IF NOT UNMAPPING VA
                                        0D51  2059
                F0 AD    56'AF      DE  0D51  2060          MOVAL   B^55$,B^MMG$L_PAGESUBR(FP) ;SKIP INIT CODE FOR SUBSEQUENT PAGES
                 00000000'GF      16  0D56  2061 55$:     JSB     G^MMG$DELPAG            ;UNMAP A PAGE OF VA SPACE
                                    05  0D5C  2062          RSB                             ;(DELPAG DOESN'T USE R8, SO ALL DONE)
                                        0D5D  2063 ;
                                        0D5D  2064 ; SPECIAL CODE FOR RESIDENT GLOBAL SECTIONS
                                        0D5D  2065 ;       THE PTE IS SIMPLY MADE VALID WITH NO WORKING SET LIST ENTRY
                                        0D5D  2066 ;       THE WINDOW BIT IS NOT SET TO AVOID CONFUSION WITH PFN-SECTIONS
                                        0D5D  2067 ;
                F0 AD    62'AF      DE  0D5D  2068 60$:     MOVAL   B^65$,B^MMG$L_PAGESUBR(FP) ;SKIP INIT CODE FOR SUBSEQUENT PAGES
                   5A    20 A5      C2  0D62  2069 65$:     SUBL    PHD$L_PSTBASOFF(R5),R10 ;SUBTRACT OFF POINTER THAT MIGHT CHANGE
                         7E    58   7D  0D66  2070          MOVQ    R8,-(SP)                ;SAVE PTE AND COUNT
             59    58    16    00    EF  0D69  2071          EXTZV   #PTE$V_GPTX,#PTE$S_GPTX,R8,R9 ;GET GPTX
```

```
59   00000000'FF49   D0  0D6E  2072          MOVL    @MMG$GL_GPTBASE[R9],R9          ;GET THE GPTE
 58   15    00   59   F0  0D76  2073          INSV    R9,#PTE$V_PFN,#PTE$S_PFN,R8 ;PUT PFN INTO PTE
        00 58    1F   E2  0D7B  2074          BBSS    #PTE$V_VALID,R8,70$     ;SET THE VALID BIT
  58   04400000 8F   CA  0D7F  2075 70$:     BICL    #PTE$M_TYP0!PTE$M_TYP1,R8
       00000000'EF   16  0D86  2076          JSB     MMG$CREPAG
       00000000'EF   16  0D8C  2077          JSB     LCKPAGTBL                ;LOCK THE PAGE TABLE
            58   8E   7D  0D92  2078          MOVQ    (SP)+,R8
               FF74   31  0D95  2079          BRW     26$                      ;GO BACK TO COMMON CODE
```

```
                     0D98  2081          .SBTTL  CHECK_WINDOW    INSURE FULLY MAPPED FILE
                     0D98  2082
                     0D98  2083 ;+
                     0D98  2084 ; Functional Description:
                     0D98  2085 ;
                     0D98  2086 ;       This routine checks that a file is completely mapped (that the
                     0D98  2087 ;       mapping pointers for the file are permanently resident) and that
                     0D98  2088 ;       the file will remain completely mapped while the section exists.
                     0D98  2089 ;
                     0D98  2090 ;       There is an assumption at work here that the ACP can tolerate the
                     0D98  2091 ;       CATHEDRAL bit being turned on while it is working on extending
                     0D98  2092 ;       the file.
                     0D98  2093 ;
                     0D98  2094 ; Calling Sequence:
                     0D98  2095 ;
                     0D98  2096 ;       BSBW    CHECK_WINDOW
                     0D98  2097 ;
                     0D98  2098 ; Input Parameters:
                     0D98  2099 ;
                     0D98  2100 ;       R6              Channel on which file is open
                     0D98  2101 ;       FP              Address of $CRMPSC impure area
                     0D98  2102 ;
                     0D98  2103 ; Implicit Input:
                     0D98  2104 ;
                     0D98  2105 ;       None
                     0D98  2106 ;
                     0D98  2107 ; Output Parameters:
                     0D98  2108 ;
                     0D98  2109 ;       R2              Channel index
                     0D98  2110 ;       R8              Address of Channel Control Block
                     0D98  2111 ;       MMG$L_EFBLK(FP) Largest block in file that can be mapped
                     0D98  2112 ;
                     0D98  2113 ; Implicit Output:
                     0D98  2114 ;
                     0D98  2115 ;       WCB$V_COMPLETE and WCB$V_CATHEDRAL bits are set in WCB$B_ACCESS.
                     0D98  2116 ;
                     0D98  2117 ; Completion Codes:
                     0D98  2118 ;
                     0D98  2119 ;       SS$_NORMAL                      ; Successful completion
                     0D98  2120 ;       SS$_EXBYTLM                     ; ACP remap operation failed
                     0D98  2121 ;       Error codes returned from IOC$VERIFYCHAN
                     0D98  2122 ;-
                     0D98  2123
                     0D98  2124 ;
                     0D98  2125 ; ***************** THE FOLLOWING CODE MAY BE PAGED ****************
                     0D98  2126 ;
                00000D98   2127          .PSECT  YF$$SYSCRMPSC
                     0D98  2128
                     0D98  2129 BEGIN_LOCKED_CODE:
                     0D98  2130
                     0D98  2131 CHECK_WINDOW:
          50    56   D0    0D98  2132          MOVL    R6,R0                   ; Input parameter to VERIFYCHAN
00000000'EF     16   0D9B  2133          JSB     IOC$VERIFYCHAN          ; Verify it and return CCB address
                     0DA1  2134
                     0DA1  2135 ; R1    Address of channel control block
                     0DA1  2136 ; R2    Index into channel table
                     0DA1  2137
```

```
                     3C   BB   0DA1  2138          PUSHR   #^M<R2,R3,R4,R5>          ; Save some registers
                   52 50  E9   0DA3  2139          BLBC    R0,25$                    ; Branch if bad channel parameter
                58   51   D0   0DA6  2140          MOVL    R1,R8                     ; Channel control block address
                50   68   D0   0DA9  2141 10$:     MOVL    CCB$L_UCB(R8),R0          ; Get UCB for device characteristics
     50   10004008 8F   38 A0   CB   0DAC  2142          BICL3   UCB$L_DEVCHAR(R0), -      ; Check that device ...
                                      0DB5  2143                  #<DEV$M_DIR!-             ;  has directories,
                                      0DB5  2144                    DEV$M_FOD!-             ;  is file oriented,
                                      0DB5  2145                    DEV$M_RND>,R0           ;  and is random access
                         44   12   0DB5  2146          BNEQ    30$                       ; Error if any characteristic is missing
           52    04 A8    01   CB   0DB7  2147          BICL3   #1,CCB$L_WIND(R8),R2    ; Get window address, clearing low bit
                         44   13   0DBC  2148          BEQL    35$                       ; Error if no file open on channel
                         15   19   0DBE  2149          BLSS    15$                       ; Branch if R2 contains window address
                   52    52   32   0DC0  2150          CVTWL   R2,R2                     ; Sign extend PSTX
           50    00000000'9F   D0   0DC3  2151          MOVL    @#CTL$GL_PHD,R0           ; Get process header address
                50    20 A0   C0   0DCA  2152          ADDL2   PHD$L_PSTBASOFF(R0),R0    ; Point R0 to base of section table
              52    0C A042   D0   0DCE  2153          MOVL    SEC$L_WINDOW(R0)[R2],R2   ; Get window address from PSTE
                         2D   18   0DD3  2154          BGEQ    35$                       ; Error unless system address
                                      0DD5  2155 15$:     SETIPL  W^LOCK_IPL                ; Lock code and synchronize
           2A 0B A2    05   E1   0DDA  2156          BBC     #WCB$V_COMPLETE,WCB$B_ACCESS(R2),40$     ; Step out of line
                                      0DDF  2157                                            ;  if file is not completely mapped
              53    18 A2   D0   0DDF  2158          MOVL    WCB$L_FCB(R2),R3          ; Get FCB address
                         09   13   0DE3  2159          BEQL    20$                       ; All done if not FCP window
                                      0DE5  2160
                                      0DE5  2161 ; Note that the previous branch relies on the fact that all non-FCP windows
                                      0DE5  2162 ; have already had the CATHEDRAL bit set. In addition, there is special
                                      0DE5  2163 ; code in routine INITSECTBL that takes care of file size for non-FCP
                                      0DE5  2164 ; files so that MMG$L_EFBLK does not have to be loaded. (In fact, the
                                      0DE5  2165 ; common exit code loads MMG$L_EFBLK with 0.)
                                      0DE5  2166
              53    3C A3   D0   0DE5  2167          MOVL    FCB$L_EFBLK(R3),R3        ; Save end-of-file block
        00 0B A2    06   E2   0DE9  2168          BBSS    #WCB$V_CATHEDRAL,WCB$B_ACCESS(R2),20$    ; Set CATHEDRAL bit
                                      0DEE  2169 20$:     SETIPL  #0                        ; Allow page faults to occur
           E4 AD    53   D0   0DF1  2170          MOVL    R3,MMG$L_EFBLK(FP)        ; Store end-of-file block
                50   01   3C   0DF5  2171          MOVZWL  #SS$_NORMAL,R0            ; Indicate success
                         3C   BA   0DF8  2172 25$:     POPR    #^M<R2,R3,R4,R5>          ; Restore registers
                         05   0DFA  2173          RSB                               ;  and return
                                      0DFB  2174
              50    01CC 8F   3C   0DFB  2175 30$:     MOVZWL  #SS$_NOTFILEDEV,R0        ; Device is not file structured
                         F6   11   0E00  2176          BRB     25$                       ; Restore registers and return
                                      0E02  2177
              50    026C 8F   3C   0E02  2178 35$:     MOVZWL  #SS$_IVCHNLSEC,R0         ; Return "invalid channel" error
                         EF   11   0E07  2179          BRB     25$                       ; Restore registers and return
                                      0E09  2180
                                      0E09  2181 ; Call ACP with REMAP control function to insure that all mapping pointers for
                                      0E09  2182 ; this file are permanently resident.
                                      0E09  2183
                                      0E09  2184 40$:     SETIPL  #0                        ; Lower IPL to call system services
           5E    00000040 8F   C2   0E0C  2185          SUBL2   #FIB$K_LENGTH,SP          ; Allocate space for FIB on stack
6E   0040 8F   00    6E   00   2C   0E13  2186          MOVC5   #0,(SP),#0,#FIB$K_LENGTH,(SP)    ; Fill FIB with zeros
             16 AE   10   B0   0E1B  2187          MOVW    #FIB$C_REMAP,FIB$W_CNTRLFUNC(SP); Set ACP control function
                         6E   DF   0E1F  2188          PUSHAL  (SP)                      ; Make FIB descriptor (address
              00000040 8F   DD   0E21  2189          PUSHL   #FIB$K_LENGTH             ;   and length)
                   55   5E   D0   0E27  2190          MOVL    SP,R5                     ; Save FIB address for $QIO call
                   54   7E   7E   0E2A  2191          MOVAQ   -(SP),R4                  ; Allocate space for IOSB
                                      0E2D  2192          $QIOW_S EFN=S^#EXE$C_SYSEFN,-
                                      0E2D  2193                  CHAN=R6,-
                                      0E2D  2194                  FUNC=#IO$_ACPCONTROL,-
```

```
                    0E2D                    IOSB=(R4),-
                    0E2D                    P1=(R5)
     51   64   3C   0E48          MOVZWL    (R4),R1                    ;SAVE STATUS FROM IOSB
  5E   50 AE   9E   0E4B          MOVAB     <FIB$K_LENGTH+8+8>(SP),SP ; AND REMOVE IOSB AND FIB FROM STACK
       A6 50   E9   0E4F          BLBC      R0,25$                     ;BRANCH IF FAILED TO QUEUE I/O REQUEST
     50   51   D0   0E52          MOVL      R1,R0                      ;OTHERWISE, GET IO STATUS IN R0
       A0 50   E9   0E55          BLBC      R0,25$                     ; AND BRANCH IF REMAP FAILED
        FF4E   31   0E58          BRW       10$                        ;REMAP WAS SUCCESSFUL, REPEAT CHECKS
                    0E5B
                    0E5B  LOCK_IPL:
        00000008    0E5B          .LONG     IPL$_SYNCH                 ; Synchronization IPL
                    0E5F
                    0E5F  END_LOCKED_CODE:
                    0E5F
                    0E5F  ; The following test insures that the code that is dynamically locked
                    0E5F  ; does not span more than two consecutive pages.
                    0E5F
                    0E5F          ASSUME    <END_LOCKED_CODE-BEGIN_LOCKED_CODE> LE 512
```

```
                                0E5F  2214          .SUBTITLE       MMG$RET_BYT_QUOTA               RETURN BYTCNT QUOTA
                                0E5F  2215 ;+
                                0E5F  2216 ;       Return BYTCNT Quota to Owner of File
                                0E5F  2217 ;
                                0E5F  2218 ; Functional Description:
                                0E5F  2219 ;
                                0E5F  2220 ;       When a window is converted to a shared WCB as part of global
                                0E5F  2221 ;       section creation, the BYTCNT quota that was charged against
                                0E5F  2222 ;       the process that opened the file must be returned.
                                0E5F  2223 ;
                                0E5F  2224 ; Input Parameter:
                                0E5F  2225 ;
                                0E5F  2226 ;       R0      Address of primary WCB
                                0E5F  2227 ;
                                0E5F  2228 ; Implicit Input:
                                0E5F  2229 ;
                                0E5F  2230 ;       JIB of process that opened file
                                0E5F  2231 ;
                                0E5F  2232 ; Output Parameters:
                                0E5F  2233 ;
                                0E5F  2234 ;       None
                                0E5F  2235 ;
                                0E5F  2236 ; Implicit Output:
                                0E5F  2237 ;
                                0E5F  2238 ;       JIB$L_BYTCNT is updated to account for the entire chain of WCBs.
                                0E5F  2239 ;
                                0E5F  2240 ;       Each WCB in a chain of WCBs has the following fields changed
                                0E5F  2241 ;
                                0E5F  2242 ;               WCB$L_PID is cleared
                                0E5F  2243 ;               WCB$W_REFCNT is set to 1
                                0E5F  2244 ;               WCB$V_SHRWCB in WCB$B_ACCESS is set
                                0E5F  2245 ;-
                                0E5F  2246
                                0E5F  2247 MMG$RET_BYT_QUOTA::
                     0F   BB    0E5F  2248          PUSHR   #^M<R0,R1,R2,R3>        ; Get some registers to work with
                                0E61  2249 5$:      DSBINT  30$                     ; Synchronize access to data
           51    0C A0   3C    0E6B  2250          MOVZWL  WCB$L_PID(R0),R1        ; Get process index
51    00000000'FF41   D0    0E6F  2251          MOVL    @SCH$GL_PCBVEC[R1],R1   ; Convert to PCB address
      0C A0    60 A1   D1    0E77  2252          CMPL    PCB$L_PID(R1),WCB$L_PID(R0)     ; Make a consistency check
                  2A   12    0E7C  2253          BNEQ    20$                     ; Return if PIDs do not match
        51    0080 C1   D0    0E7E  2254          MOVL    PCB$L_JIB(R1),R1        ; Finally, store JIB address
                  53   D4    0E83  2255          CLRL    R3                      ; Initialize sum
                               0E85  2256
           52    08 A0   3C    0E85  2257 10$:     MOVZWL  WCB$W_SIZE(R0),R2       ; Extract size of next WCB
                 53   52   C0    0E89  2258          ADDL2   R2,R3                   ; Include in sum
                    0C A0   D4    0E8C  2259          CLRL    WCB$L_PID(R0)           ; Zap PID to eliminate link to process
                0E A0   01   B0    0E8F  2260          MOVW    #1,WCB$W_REFCNT(R0)     ; Initialize REFCNT for one access
                0B A0   08   88    0E93  2261          BISB    #WCB$M_SHRWCB,WCB$B_ACCESS(R0)  ; Set SHRWCB bit
                50   20 A0   D0    0E97  2262          MOVL    WCB$L_LINK(R0),R0       ; Get next WCB in chain
                  E8   12    0E9B  2263          BNEQ    10$                     ; Loop back if next WCB exists
                               0E9D  2264
                20 A1   53   C0    0E9D  2265          ADDL2   R3,JIB$L_BYTCNT(R1)     ; Restore BYTCNT quota
                24 A1   53   C0    0EA1  2266          ADDL2   R3,JIB$L_BYTLM(R1)      ; Restore byte limit also.
                      30 A1   B6    0EA5  2267          INCW    JIB$W_FILCNT(R1)        ; One less file to worry about
                               0EA8  2268 20$:     ENBINT                          ; Allow rescheduling to occur
                     0F   BA    0EAB  2269          POPR    #^M<R0,R1,R2,R3>        ; Restore registers
                          05   0EAD  2270          RSB                             ;  and return
```

```
                     0EAE  2271
00000008             0EAE  2272 30$:    .LONG   IPL$_SYNCH              ; Synchronization IPL
                     0EB2  2273
                     0EB2  2274          ASSUME  <.-5$> LE 512           ; Can only lock 512 bytes this way
                     0EB2  2275
```

```
                                0EB2          .SUBTITLE       READ_RESIDENT  -  INITIALIZE A RESIDENT GLOBAL SECTION
                                0EB2  ;+
                                0EB2  ;
                                0EB2  ; Functional Description:
                                0EB2  ;
                                0EB2  ;       The contents of a resident global section is read into memory.
                                0EB2  ;       Internal interfaces are used to queue the IRP.
                                0EB2  ;       On error, the physical pages are released.
                                0EB2  ;
                                0EB2  ; Input Parameters
                                0EB2  ;
                                0EB2  ;       R3 - First global PTE
                                0EB2  ;       R7 - page count
                                0EB2  ;
                                0EB2  ; Registers destroyed
                                0EB2  ;
                                0EB2  ;       R0,R1,R2,R3,R4,R7
                                0EB2  ;
                                0EB2  ;*******************************************************************
                                0EB2  ;*********LESS THAN IDEAL FOR GENERAL USE***************************
                                0EB2  ; THIS ROUTINE DOES A WAIT FOR IO COMPLETION HOLDING GSD MUTEX
                                0EB2  ; THIS ROUTINE DOES NOT CHECK DIOCNT
                                0EB2  ; THIS IS NOT ACCEPTABLE FOR A DOCUMENTED, GENERAL USE INTERFACE
                                0EB2  ; THIS IS INTENDED ONLY FOR USE DURING SYSTEM INITIALIZATION
                                0EB2  ;*******************************************************************
                                0EB2  ;-
                                0EB2
                                0EB2  READ_RESIDENT:
                    57    DD    0EB2          PUSHL   R7                      ; Save count
          51   00C4 8F    3C    0EB4          MOVZWL  #IRP$C_LENGTH,R1        ; Get an IRP
          00000000'EF     16    0EB9          JSB     EXE$ALONONPAGED         ; Return IRP in R2
                 6D 50    E9    0EBF          BLBC    R0,50$                  ; None available
     54   00000000'EF     D0    0EC2          MOVL    SCH$GL_CURPCB,R4
                    28    BB    0EC9          PUSHR   #^M<R3,R5>
                55  52    D0    0ECB          MOVL    R2,R5                   ; IRP address
  14 A5   00000F68'EF     DE    0ECE          MOVAL   200$,IRP$L_ASTPRM(R5)   ; AST to free the IRP
        23 A5    2F A4    90    0ED6          MOVB    PCB$B_PRIB(R4),IRP$B_PRI(R5) ; Transfer priority
                 38 A5    D4    0EDB          CLRL    IRP$L_IOST1(R5)
          50    10 AA     D0    0EDE          MOVL    SEC$L_VBN(R10),R0       ; Starting VBN
          51    57  09    78    0EE2          ASHL    #9,R7,R1                ; Byte count
          52    0C AA     D0    0EE6          MOVL    SEC$L_WINDOW(R10),R2    ; WCB address
                57  55    D0    0EEA          MOVL    R5,R7                   ; Save IRP address
          00000000'EF     16    0EED          JSB     EXE$BLDPKTSWPR  ; Build and queue IRP
                    28    BA    0EF3          POPR    #^M<R3,R5>              ; Get back PHD, SVAPTE
                51  01    D0    0EF5  10$:    MOVL    #RSN$_ASTWAIT,R1        ; Wait for an AST
          00000000'EF     16    0EF8          JSB     MMG$RESRCWAIT           ; Set up to wait for the resource
     12   00000F64'EF     DA    0EFE          MTPR    100$,#PR$_IPL
                    7E    DC    0F05          MOVPSL  -(SP)                   ; Wait requires PC-PSL
          00000000'EF     16    0F07          JSB     MMG$SVPCTX              ; Go wait
                                0F0D  ;
                                0F0D  ;       We need some way to tell when the I/O is done, but we can't get an
                                0F0D  ;       AST since we are at ASTDEL.  Instead, look for the IRP to be put
                                0F0D  ;       on the AST queue.
                                0F0D  ;
          50    10 A4     9E    0F0D          MOVAB   PCB$L_ASTQFL(R4),R0     ; Get AST queue
                51  50    D0    0F11          MOVL    R0,R1                   ; Save it for comparison
                50  60    D0    0F14  20$:    MOVL    (R0),R0                 ; Next AST on queue
```

```
                51    50    D1  0F17  2334          CMPL    R0,R1                   ; Are we back at beginning
                      D9    13  0F1A  2335          BEQL    10$                     ; Yes - go wait some more
                57    50    D1  0F1C  2336          CMPL    R0,R7                   ; Is this the one we want
                      F3    12  0F1F  2337          BNEQ    20$                     ; No - look for another one
                12    02    DA  0F21  2338          MTPR    #IPL$_ASTDEL,#PR$_IPL   ; Go back to reasonable IPL
                                0F24  2339 ;
              50    38 A7    3C  0F24  2340          MOVZWL  IRP$L_IOST1(R7),R0      ; Get IO status
                 04 50    E9  0F28  2341          BLBC    R0,50$                  ; Error
                5E    04    C0  0F2B  2342          ADDL    #4,SP                   ; Clean up stack
                            05  0F2E  2343          RSB
                                0F2F  2344
                                0F2F  2345 ;
                                0F2F  2346 ; Error - clean up and release physical memory
                                0F2F  2347 ;
                57 8ED0  0F2F  2348 50$:    POPL    R7                      ; Get back the page count
                      50    DD  0F32  2349          PUSHL   R0                      ; Save the error status
          12  00000F64'EF    DA  0F34  2350 60$:    MTPR    100$,#PR$_IPL
                      50    63  D0  0F3B  2351          MOVL    (R3),R0                 ; Get the GPTE
          50  FFE00000 8F    CA  0F3E  2352          BICL    #^C<PTE$M_PFN>,R0       ; Leave only PFN
             00000000'FF40    B7  0F45  2353          DECW    @PFN$AW_REFCNT[R0]
               00000000'EF    16  0F4C  2354          JSB     MMG$RELPFN              ; Release the page
               00000000'EF    16  0F52  2355          JSB     MMG$DECPTREF            ; Global page table ref count
                      53    04  C0  0F58  2356          ADDL    #4,R3                   ; Next GPTE
                      12    02  DA  0F5B  2357          MTPR    #IPL$_ASTDEL,#PR$_IPL   ; Go back to reasonable IPL
                        D3 57    F5  0F5E  2358          SOBGTR  R7,60$
                         F47B    31  0F61  2359          BRW     FRESTXERR               ; Error clean-up - expects R0 on stack
                                0F64  2360
                      00000008  0F64  2361 100$:    .LONG   IPL$_SYNCH              ; Synchronization IPL
                                0F68  2362          ASSUME  <. - 50$> LE 512        ; Can only lock 512 bytes this way
                                0F68  2363
                                0F68  2364 ;
                                0F68  2365 ; I/O completion AST - delivered after IPL drops
                                0F68  2366 ; Free the IRP
                                0F68  2367 ;
                   50    55    D0  0F68  2368 200$:    MOVL    R5,R0
             00000000'EF    16  0F6B  2369          JSB     EXE$DEANONPAGED
                            05  0F71  2370          RSB
```

```
                     0F72  2372 ;+
                     0F72  2373 ;
                     0F72  2374 ; Functional description:
                     0F72  2375 ;
                     0F72  2376 ;       Lock the page table in memory.
                     0F72  2377 ;
                     0F72  2378 ; Input Parameters
                     0F72  2379 ;
                     0F72  2380 ;    For LCKPAGTBL only
                     0F72  2381 ;       R2 - VA of page represented by page table to be locked
                     0F72  2382 ;
                     0F72  2383 ;    For LCKPAGTBL1 only
                     0F72  2384 ;       R1 - virtual page number
                     0F72  2385 ;       R3 - pointer to base of appropriate page table
                     0F72  2386 ;
                     0F72  2387 ; Registers destroyed
                     0F72  2388 ;       LCKPAGTBL1 only
                     0F72  2389 ;        R2
                     0F72  2390 ;
                     0F72  2391 ;
                     0F72  2392 ;-
                     0F72  2393 ; ****************************************************************************
                     0F72  2394 ;************** THIS CODE MUST NOT PAGE ************************************
                     0F72  2395 ;
                     0F72  2396         .SAVE_PSECT
                00000000   2397         .PSECT $MMGCOD,LONG
                     0000  2398 ;
                     0000  2399 ;****************************************************************************
                     0000  2400 ;
                     0000  2401 LCKPAGTBL:
                       0B  BB   0000  2402         PUSHR   #^M<R0,R1,R3>
                   FFFB'   30   0002  2403         BSBW    MMG$PTEREF              ;GET THE SVAPTE
                50    01   D0   0005  2404         MOVL    #1,R0                   ;LOCK INDICATOR
                   FFF5'   30   0008  2405         BSBW    MMG$MOVPTLOCK           ;DO ALL THE WORK
                12    02   DA   000B  2406         MTPR    #IPL$_ASTDEL,#PR$_IPL   ;WE DON'T WANT TO STAY AT SYNCH
                      0B   BA   000E  2407         POPR    #^M<R0,R1,R3>
                      05   0010  2408         RSB
                           0011  2409
                           0011  2410 LCKPAGTBL1:
                       09   BB   0011  2411         PUSHR   #^M<R0,R3>
                     29 59  E8   0013  2412         BLBS    R9,20$                  ;Indicator for first time through loop
                51    7F  8F   93   0016  2413         BITB    #^X7F,R1                ;Is it first one in new page
                      23   13   001A  2414         BEQL    20$                     ;Yes - do it the hard way
                12    08   DA   001C  2415         MTPR    #IPL$_SYNCH,#PR$_IPL    ;Don't let anything change
             53    00 B341   DE   001F  2416         MOVAL   @(R3)[R1],R3            ;Get the SVAPTE
          52 53    00C8 C5   C3   0024  2417         SUBL3   PHD$L_POBR(R5),R3,R2    ;Byte offset of PTE
       50    52      F7  8F   78   002A  2418         ASHL    #-9,R2,R0               ;Byte index of containing page table
       52    55      64  A5   C1   002F  2419         ADDL3   PHD$L_PTWSLELCK(R5),R5,R2 ;Locked working set list entries
                52    50   C0   0034  2420         ADDL2   R0,R2                   ;Address of count byte for # of locked
                           0037  2421                                                 ; WSLE's in the page table
                      62   96   0037  2422         INCB    (R2)                    ;Add one for WINDOW/MA780 global page
                12    02   DA   0039  2423 10$:     MTPR    #IPL$_ASTDEL,#PR$_IPL   ;We don't want to stay at synch
                      09   BA   003C  2424         POPR    #^M<R0,R3>
                      05   003E  2425         RSB
                           003F  2426 ;
                59    01   8A   003F  2427 20$:     BICB    #1,R9                   ;Just in case this is the first time
                00 B341    D5   0042  2428         TSTL    @(R3)[R1]               ;Make sure page table is resident
```

```
        12   08   DA  0046  2429        MTPR    #IPL$_SYNCH,#PR$_IPL     ;Don't let anything change
   53   00 B341   DE  0049  2430        MOVAL   @(R3)[R1],R3             ;Get the SVAPTE
        50   01   D0  004E  2431        MOVL    #1,R0                    ;Lock indicator
           FFAC'  30  0051  2432        BSBW    MMG$MOVPTLOCK            ;Do all the work
             E3   11  0054  2433        BRB     10$
                  00000F72  2434        .RESTORE_PSECT
                      0F72  2435
                      0F72  2436        .END
```

```
$$T1                            =  00000001
ACMODE                          =  0000000C
ARB$L_RIGHTSLIST                =  00000020
ARB$Q_PRIV                      =  00000000
ARM$M_EXECUTE                   =  00000004
ARM$M_READ                      =  00000001
ARM$M_WRITE                     =  00000002
BADMATCHCTL                        000002B2 R      02
BEGIN_LOCKED_CODE                  00000D98 R      02
BRGSTFULL                          000002E3 R      02
BRW_70$                            000002A9 R      02
CCB$L_UCB                       =  00000000
CCB$L_WIND                      =  00000004
CCB$W_IOC                       =  0000000A
CHAN                            =  00000020
CHECK_WINDOW                       00000D98 R      02
COMMON_INIT                        0000022F R      02
COMMON_INIT1                       0000014C R      02
COMMON_MAP                         00000653 R      02
CRMPSC_ACCVIO                      00000012 R      02
CRMPSC_FOUND                       00000059 R      02
CRMPSC_NOINADR                     00000000 R      02
CRMPSC_NOPRIV                      000000E3 R      02
CRMPSC_PROCESS                     000000D9 R      02
CRMPSC_RET                         00000015 R      02
CTL$GL_PHD                         ********   X    02
DEV$M_DIR                       =  00000008
DEV$M_FOD                       =  00004000
DEV$M_RND                       =  10000000
DYN$C_EXTGSD                    =  00000028
DYN$C_GSD                       =  00000015
DYN$C_SHMGSD                    =  00000029
END_LOCKED_CODE                    00000E5F R      02
EXE$ALLOCATE                       ********   X    02
EXE$ALONONPAGED                    ********   X    02
EXE$ALOPAGED                       ********   X    02
EXE$BLDPKTSWPR                     ********   X    02
EXE$CHECKPROT_16                   ********   X    02
EXE$CRMPSC                         00000000 RG     03
EXE$C_SYSEFN                       ********   X    02
EXE$DEANONPAGED                    ********   X    02
EXE$DEAPAGED                       ********   X    02
EXE$GL_GPT                         ********   X    02
EXE$GL_GSDGRPFL                    ********   X    02
EXE$GL_GSDMTX                      ********   X    02
EXE$MGBLSC                         00000008 RG     03
EXE_CRMPSC                         00000016 R      02
EXE_MGBLSC                         00000422 R      02
FAST_MAP                           00000874 R      02
FCB$C_EFBLK                     =  0000003C
FCB$L_FILEOWNER                 =  00000058
FCB$W_FILEPROT                  =  00000070
FCB$W_TCNT                      =  00000020
FIB$C_REMAP                     =  00000010
FIB$K_LENGTH                    =  00000040
FIB$W_CNTRLFUNC                 =  00000016
FLAGS                           =  00000010
FLAG_ERROR                         000000DC R      02
FLAG_ERROR1                        0000000B R      02
FRESTXERR                          000003DF R      02
GET_READ_ACCESS                    00000A00 R      02
GPTFULL                            000003D9 R      02
GSD$B_HASH                      =  0000000B
GSD$B_TYPE                      =  0000000A
GSD$C_EXTGSDLNG                 =  00000031
GSD$C_LENGTH                    =  00000023
GSD$C_PFNBASMAX                 =  00000004
GSD$L_BASEPFN                   =  00000024
GSD$L_BASPFN1                   =  00000054
GSD$L_FILUIC                    =  00000010
GSD$L_GSDBL                     =  00000004
GSD$L_GSDFL                     =  00000000
GSD$L_IDENT                     =  00000018
GSD$L_PAGES                     =  00000028
GSD$L_PCBUIC                    =  0000000C
GSD$L_REFCNT                    =  0000002C
GSD$T_GSDNAM                    =  00000022
GSD$T_PFNGSDNAM                 =  00000030
GSD$V_DUPGSD                    =  00000004
GSD$V_INITFAIL                  =  00000003
GSD$V_LOCKED                    =  00000001
GSD$V_VALID                     =  00000000
GSD$W_FLAGS                     =  00000020
GSD$W_GSTX                      =  00000016
GSD$W_PROT                      =  00000014
GSD$W_SIZE                      =  00000008
GSDNAM                          =  00000014
GSTFULL                            00000404 R      02
GSTFULL1                           000002B7 R      02
IDENT                           =  00000018
ILL_PAG_CNT                        000000E9 R      02
ILL_RELPAG                         00000613 R      02
INADR                           =  00000004
INADRERR                           00000421 R      02
INITSECTBL                         00000A5D R      02
IO$_ACPCONTROL                  =  00000038
IOC$VERIFYCHAN                     ********   X    02
IPL$_ASTDEL                     =  00000002
IPL$_SYNCH                      =  00000008
IRP$B_PRI                       =  00000023
IRP$C_LENGTH                    =  000000C4
IRP$L_ASTPRM                    =  00000014
IRP$L_IOST1                     =  00000038
JIB$L_BYTCNT                    =  00000020
JIB$L_BYTLM                     =  00000024
JIB$W_FILCNT                    =  00000030
LCKPAGTBL                          00000000 R      04
LCKPAGTBL1                         00000011 R      04
LOCAL_MEM_GSD                      00000152 R      02
LOCK_IPL                           00000E5B R      02
MAPGBLSEC1                         0000047D R      02
MAPGBLSEC2                         000004B2 R      02
MAPGBLSEC3                         000004BB R      02
MAPGBLSEC4                         0000052F R      02
```

```
MAPSEC_RET                      00000420 R      02
MAP_LOCAL_MEM                   00000634 R      02
MAP_NXT_BASE                    0000053E R      02
MAP_PROCESS_SEC                 00000B6B R      02
MAP_TO_FILE                     00000259 R      02
MGB[SC_NOPRIV                   000004EB R      02
MMG$ALCSTX                      ********    X   02
MMG$ALLOCPFN                    ********    X   02
MMG$ALOSHMGSD                   ********    X   02
MMG$ALOSHMPAG                   ********    X   02
MMG$CREDEL                      ********    X   02
MMG$CREPAG                      ********    X   02
MMG$C_LENGTH                  = FFFFFFE4
MMG$DALCSTX                     ********    X   02
MMG$DALCSTXSCN                  ********    X   02
MMG$DALCSTXSCN1                 ********    X   02
MMG$DECPTREF                    ********    X   02
MMG$DECSECREF                   ********    X   02
MMG$DECSHMREF                   ********    X   02
MMG$DELGBLWCB                   ********    X   02
MMG$DELPAG                      ********    X   02
MMG$FAST_CREATE                 ********    X   02
MMG$FINDGSNOTRN                 ********    X   02
MMG$GL_GBLPAGFIL                ********    X   02
MMG$GL_GPTBASE                  ********    X   02
MMG$GL_MAXMEM                   ********    X   02
MMG$GL_SYSPHD                   ********    X   02
MMG$GSDMTXULK                   ********    X   02
MMG$GSDSCN                      ********    X   02
MMG$INADRINI                    ********    X   02
MMG$INCPTREF                    ********    X   02
MMG$INCSHMREF                   ********    X   02
MMG$IN_REGION                   ********    X   02
MMG$L_CALLEDIPL               = FFFFFFF8
MMG$L_EFBLK                   = FFFFFFE4
MMG$L_MAXACMODE               = FFFFFFFC
MMG$L_PAGESUBR                = FFFFFFF0
MMG$L_SAVRETADR               = FFFFFFF4
MMG$L_VFYFLAGS                = FFFFFFE8
MMG$MAPSECPAG                   00000CE1 R      02
MMG$MOVPTLOCK                   ********    X   04
MMG$PGFLTWAIT                   ********    X   02
MMG$PTEREF                      ********    X   04
MMG$READ_GSD                    ********    X   02
MMG$RELPFN                      ********    X   02
MMG$RESRCWAIT                   ********    X   02
MMG$RETADRINI                   ********    X   02
MMG$RETRANGE                    ********    X   02
MMG$RET_BYT_QUOTA               00000E5F RG     02
MMG$SET_BITMAP                  ********    X   02
MMG$SHMTXLK                     ********    X   02
MMG$SHMTXULK                    ********    X   02
MMG$SVPCTX                      ********    X   02
MMG$TRY_ALL                     ********    X   02
MMG$UNIQUEGSD                   ********    X   02
MMG$VFYSECFLG                   ********    X   02
MMG$V_CHGPAGFIL               = 00000008

MMG$V_NOWAIT_IPL0             = 0000000A
MOVGSDNAM                       00000172 R      02
NORMAL_GSD                      000001CA R      02
NO_FILE_INIT                    00000222 R      02
NO_MORE_PAGES                   00000784 R      02
NO_WRT_ACCESS                   000009FD R      02
PCB$B_PRIB                    = 0000002F
PCB$L_ARB                     = 0000008C
PCB$L_ASTQFL                  = 00000010
PCB$L_JIB                     = 00000080
PCB$L_PID                     = 00000060
PCB$L_UIC                     = 000000BC
PFC                           = 00000030
PFN                           = 00000028
PFN$AB_STATE                    ********    X   02
PFN$AB_TYPE                     ********    X   02
PFN$AL_BAK                      ********    X   02
PFN$AL_PTE                      ********    X   02
PFN$AW_REFCNT                   ********    X   02
PFN$C_ACTIVE                  = 00000007
PFN$C_GLOBAL                  = 00000002
PFNMAP_GSD                      000001E8 R      02
PFNMAP_GSD_1                    0000014F R      02
PHD$L_FREP0VA                 = 00000028
PHD$L_FREP1VA                 = 00000030
PHD$L_MPINHIBIT               = 00000104
PHD$L_P0BR                    = 000000C8
PHD$L_P1BR                    = 000000D0
PHD$L_PSTBASOFF               = 00000020
PHD$L_PTWSLELCK               = 00000064
PHD$Q_PRIVMSK                 = 00000000
PHD$V_DALCSTX                 = 00000001
PHD$W_FLAGS                   = 00000036
PIECE_MAPPED                    0000073C R      02
PR$_IPL                       = 00000012
PRE_MAP_ERR                     00000721 R      02
PRIPL                           000003D5 R      02
PROT                          = 0000002C
PRT$C_UW                      = 00000004
PRV$V_CMKRNL                  = 00000000
PRV$V_PFNMAP                  = 0000001A
PRV$V_PRMGBL                  = 00000018
PRV$V_SHMEM                   = 0000001B
PRV$V_SYSGBL                  = 00000019
PSL$V_PRVMOD                  = 00000016
PTE$M_OWN                     = 01800000
PTE$M_PFN                     = 001FFFFF
PTE$M_PROT                    = 78000000
PTE$M_TYP0                    = 00400000
PTE$M_TYP1                    = 04000000
PTE$M_VALID                   = 80000000
PTE$M_WINDOW                  = 00200000
PTE$S_GPTX                    = 00000016
PTE$S_OWN                     = 00000002
PTE$S_PFN                     = 00000015
PTE$S_PROT                    = 00000004
PTE$V_GPTX                    = 00000000
```

```
PTE$V_OWN                       =  00000017
PTE$V_PFN                       =  00000000
PTE$V_PROT                      =  0000001B
PTE$V_TYPO                      =  00000016
PTE$V_VALID                     =  0000001F
PTE$V_WINDOW                    =  00000015
READ_RESIDENT                      00000EB2  R      02
RELEAS_GSD_RET                     000001AA  R      02
RELEAS_PAG_RET                     00000193  R      02
RELEAS_SHMGS                       000007F2  R      02
RELPAG                          =  0000001C
RETADR                          =  00000008
RSN$_ASTWAIT                    =  00000001
SCH$GL_CURPCB                      ********    X    02
SCH$GL_PCBVEC                      ********    X    02
SCH$GQ_FPGWQ                       ********    X    02
SCH$LOCKW                          ********    X    02
SEC$B_PFC                       =  0000000B
SEC$C_LENGTH                    =  00000020
SEC$K_MATLEQ                    =  00000002
SEC$L_GSD                       =  00000000
SEC$L_PAGCNT                    =  0000001C
SEC$L_REFCNT                    =  00000018
SEC$L_VBN                       =  00000010
SEC$L_VPXPFC                    =  00000C08
SEC$L_WINDOW                    =  0000000C
SEC$M_CRF                       =  00000002
SEC$M_DZRO                      =  00000004
SEC$M_WRT                       =  00000008
SEC$S_AMOD                      =  00000002
SEC$S_WRTMOD                    =  00000002
SEC$V_AMOD                      =  00000008
SEC$V_CRF                       =  00000001
SEC$V_DZRO                      =  00000002
SEC$V_EXECUTE                   =  00000014
SEC$V_EXPREG                    =  00000011
SEC$V_GBL                       =  00000000
SEC$V_PAGFIL                    =  00000013
SEC$V_PERM                      =  0000000E
SEC$V_PFNMAP                    =  00000010
SEC$V_RESIDENT                  =  0000000D
SEC$V_SHMGS                     =  00000004
SEC$V_SYSGBL                    =  0000000F
SEC$V_WRT                       =  00000003
SEC$V_WRTMOD                    =  00000006
SEC$W_FLAGS                     =  00000014
SEC$W_SECXBL                    =  00000006
SEC$W_SECXFL                    =  00000004
SECPAGCNT                       =  00000024
SETSECPROTOWN                      00000996  R      02
SET_PTE_PROT                       00000A05  R      02
SGN$GL_PTPAGCNT                    ********    X    02
SHB$B_PORT                      =  00000015
SHB$L_BASGSPFN                  =  00000010
SHB$L_DATAPAGE                  =  00000004
SHB$L_REFCNT                    =  0000000C
SHD$B_FLAGS                     =  0000009F
SHD$V_BITMAPLCK                 =  00000001
SHD$W_GSDQUOTA                  =  0000003C
SHM_GSD                            00000601  R      02
SHM_GS_MAP                         000002AC  R      02
SHM_NOT_MAP                        000000EF  R      02
SHM_UNMAPPED                       0000080C  R      02
SS$_ABORT                       =  0000002C
SS$_ACCVIO                      =  0000000C
SS$_CREATED                     =  00000619
SS$_ENDOFFILE                   =  00000870
SS$_EXGBLPAGFIL                 =  00002164
SS$_GPTFULL                     =  000000C4
SS$_GSDFULL                     =  000000CC
SS$_IDMISMATCH                  =  000003F4
SS$_ILLPAGCNT                   =  000000FC
SS$_IVCHNLSEC                   =  0000026C
SS$_IVLVEC                      =  0000203C
SS$_IVSECFLG                    =  0000016C
SS$_IVSECIDCTL                  =  000002E4
SS$_IVSSRQ                      =  00000174
SS$_NOPRIV                      =  00000024
SS$_NORMAL                      =  00000001
SS$_NOSUCHSEC                   =  00000978
SS$_NOTFILEDEV                  =  000001CC
SS$_NOWRT                       =  000003FC
SS$_SHMGSNOTMAP                 =  0000036C
SYS$QIOW                           ********  GX    02
TRY_NXT_BASE                       0000061B  R      02
UCB$L_DEVCHAR                   =  00000038
ULKGSDMTXRET                       00000411  R      02
ULKGSDMTXRET1                      000000E6  R      02
ULKGSDMTXRET2                      00000413  R      02
VA$S_VPN                        =  00000015
VA$V_P1                         =  0000001E
VA$V_VPN                        =  00000009
VBN                             =  00000028
WCB$B_ACCESS                    =  0000000B
WCB$L_FCB                       =  00000018
WCB$L_LINK                      =  00000020
WCB$L_PID                       =  0000000C
WCB$M_SHRWCB                    =  00000008
WCB$V_CATHEDRAL                 =  00000006
WCB$V_COMPLETE                  =  00000005
WCB$V_NOTRUNC                   =  0000000B
WCB$V_SHRWCB                    =  00000003
WCB$V_WRITE                     =  00000001
WCB$W_ACON                      =  00000014
WCB$W_NMAP                      =  00000016
WCB$W_P1_COUNT                  =  00000030
WCB$W_REFCNT                    =  0000000E
WCB$W_SIZE                      =  00000008
WRTMOD_TBL                         000009FB  R      02
```

E 5

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YF$$SYSCRMPSC	00000F72 (3954.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
Y$EXEPAGED	00000010 (16.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
$MMGCOD	00000056 (86.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.05	00:00:01.29
Command processing	124	00:00:00.53	00:00:05.56
Pass 1	695	00:00:30.80	00:01:38.85
Symbol table sort	0	00:00:04.67	00:00:15.89
Pass 2	411	00:00:07.61	00:00:28.35
Symbol table output	1	00:00:00.26	00:00:00.91
Psect synopsis output	0	00:00:00.03	00:00:00.49
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1268	00:00:43.96	00:02:31.35

The working set limit was 2400 pages.
181708 bytes (355 pages) of virtual memory were used to buffer the intermediate code.
There were 160 pages of symbol table space allocated to hold 2875 non-local and 203 local symbols.
2436 source lines were read in Pass 1, producing 36 object records in Pass 2.
47 pages of virtual memory were used to define 46 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	25
_255DUA28:[SYSLIB]STARLET.MLB;2	18
TOTALS (all libraries)	43

2991 GETS were required to define 43 macros.

There were no errors, warnings or information messages.

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SYSCRMPSC
LIS

SYSDCLEXH
LIS

SYSDEVALC
LIS

SYSCVTIIM
LIS

SYSDGBLSC
LIS

SYSENQDEQ
LIS

SYSDCLCMH
LIS

SYSDERLMB
LIS

SYSDASSGN
LIS

SYSDELPRC
LIS