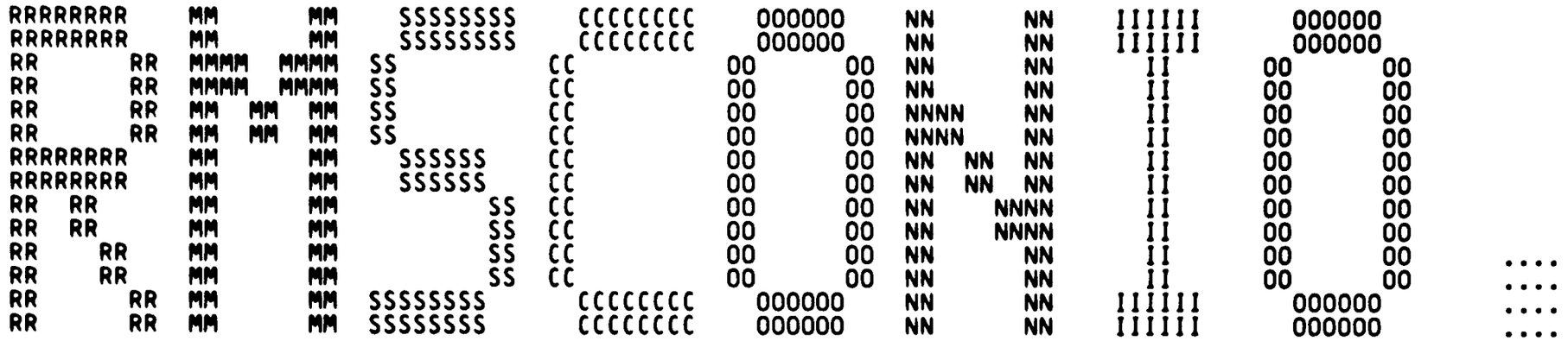
```
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO    TTTTTTTTTTTTTTT     SSSSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT         SSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBBBBBBBBBBB     OOO         OOO  OOO         OOO       TTT            SSSSSSSSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBB         BBB  OOO         OOO  OOO         OOO       TTT                     SSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
BBBBBBBBBBBB        OOOOOOOOO        OOOOOOOOO         TTT         SSSSSSSSSSS
```


RRRRRRRR MM MM SSSSSSSS CCCCCCCC 000000 NN NN IIIIII 000000
RRRRRRRR MM MM SSSSSSSS CCCCCCCC 000000 NN NN IIIIII 000000
RR RR MMMM MMMM SS CC 00 00 NN NN II 00 00
RR RR MMMM MMMM SS CC 00 00 NN NN II 00 00
RR RR MM MM MM SS CC 00 00 NNNN NN II 00 00
RR RR MM MM MM SS CC 00 00 NNNN NN II 00 00
RRRRRRRR MM MM SSSSSS CC 00 00 NN NN NN II 00 00
RRRRRRRR MM MM SSSSSS CC 00 00 NN NN NN II 00 00
RR RR MM MM SS CC 00 00 NN NNNN II 00 00
RR RR MM MM SS CC 00 00 NN NNNN II 00 00
RR RR MM MM SS CC 00 00 NN NN II 00 00
RR RR MM MM SS CC 00 00 NN NN II 00 00
RR RR MM MM SSSSSSSS CCCCCCCC 000000 NN NN IIIIII 000000
RR RR MM MM SSSSSSSS CCCCCCCC 000000 NN NN IIIIII 000000

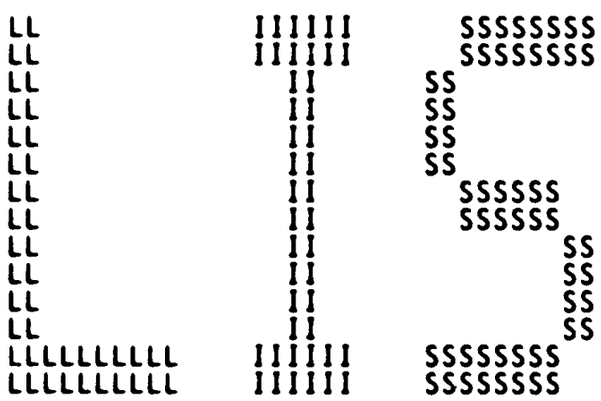

LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

Table of contents

```
0000     1          .TITLE  RMSCONIO - RMS INPUT/OUTPUT MODULE
0000     2          .IDENT  'V04-000'
0000     3          .DEFAULT DISPLACEMENT,LONG
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY:     SYSGEN UTILITY
0000    31 ;
0000    32 ; ABSTRACT:     RMSCONIO PROVIDES ROUTINES TO PERFORM CONSOLE, COMMAND
0000    33 ;               AND FILE I/O FOR THE SYSGEN UTILITY
0000    34 ;
0000    35 ; ENVIRONMENT:  NATIVE/USER MODE CODE
0000    36 ;
0000    37 ; AUTHOR:       R.I. HUSTVEDT, CREATION DATE: 7-SEP-1977
0000    38 ;
0000    39 ; MODIFIED BY:
0000    40 ;
0000    41 ;       V03-018 WHM0007         Bill Matthews           27-Jun-1984
0000    42 ;               Don't set/clear BOOCMD$V_TERMINAL for /OUTPUT=.
0000    43 ;
0000    44 ;       V03-017 WHM0006         Bill Matthews           26-Jun-1984
0000    45 ;               Added RIO_OUTFAB3 and RIO_OUTRAB3 for use by SET/OUTPUT so it
0000    46 ;               won't conflict with /OUTPOT=.
0000    47 ;
0000    48 ;       V03-016 WHM0005         Bill Matthews           25-Apr-1984
0000    49 ;               Added an initial allocation size for parameter file of 15 blocks.
0000    50 ;
0000    51 ;       V03-015 WHM0004         Bill Matthews           04-Apr-1984
0000    52 ;               Added routine BOO$WRTSYSPARFILE to support the writing of
0000    53 ;               the default system parameter file.
0000    54 ;               Added support for writing ascii sysgen parameters that are
0000    55 ;               longer than 4 characters to a parameter file.
0000    56 ;
0000    57 ;       V03-014 WHM0003         Bill Matthews           28-Feb-1984
```

```
0000     58 ;                Make BOO$UFOOPEN use SYS$SYSTEM:.EXE as a default name string.
0000     59 ;
0000     60 ;        V03-013 WMC0013         Wayne Cardoza           16-Jan-1983
0000     61 ;                Remove or bypass checkpoint related code.
0000     62 ;
0000     63 ;        V03-012 SOP0001         J. R. Sopka             27 December 1983
0000     64 ;                Invoke MMG$RET_BYT_QUOTA during processing to convert WCB
0000     65 ;                for page or swap file to a permanent system data structure.
0000     66 ;
0000     67 ;        V03-011 WHM0002         Bill Matthews           09-Dec-1983
0000     68 ;                Modified BOO$CREATE to set the CBT bit in the
0000     69 ;                allocation XAB before an RMS $EXTEND is attempted.
0000     70 ;
0000     71 ;        V03-010 WHM0001         Bill Matthews           17-Nov-1983
0000     72 ;                Modified BOO$INSTALL to deassign the channel to the file
0000     73 ;                on successful installation of the new swap or page file
0000     74 ;
0000     75 ;        V03-009 WMC0004         Wayne Cardoza           27-Jul-1983
0000     76 ;                Add BOO$FILCLOSE, BOO$UFOOPEN.
0000     77 ;
0000     78 ;        V03-008 MSH0002         Maryann Hinden          24-Jun-1983
0000     79 ;                Change $BOODEF to $BOOCMDDEF.
0000     80 ;
0000     81 ;        V03-007 MSH0001         Maryann Hinden          13-Jun-1983
0000     82 ;                Use $BOODEF from BOOTS.MLB.
0000     83 ;
0000     84 ;        V03-006 WMC0003         Wayne Cardoza           10-Mar-1983
0000     85 ;                If page file is not installed for checkpointing, invalidate
0000     86 ;                any saved checkpoints.
0000     87 ;
0000     88 ;        V03-005 WMC0002         Wayne Cardoza           15-Feb-1983
0000     89 ;                Don't leave file locked or pool in use if checkpoint
0000     90 ;                portion of page file installation fails.
0000     91 ;
0000     92 ;        V03-004 WMC0001         Wayne Cardoza           12-Aug-1982
0000     93 ;                Add support for checkpointing in page file installation.
0000     94 ;
0000     95 ;        V03-003 JLV0204         Jake VanNoy             29-MAR-1982
0000     96 ;                Set UFO bit in SWAPFHCFAB in INSTALL.
0000     97 ;
0000     98 ;        V03-002 JLV0198         Jake VanNoy             17-MAR-1982
0000     99 ;                New create logic to extend existing files if /CONTIGUOUS
0000    100 ;                isn't specified.
0000    101 ;
0000    102 ;        V03-001 LJK0148         Lawrence J. Kenah       16-Mar-1982
0000    103 ;                Remove ACP call for partially mapped file as that situation
0000    104 ;                can never happen.
0000    105 ;
0000    106 ;--
0000    107         .SBTTL  DECLARATIONS
0000    108 ;
0000    109 ; INCLUDE FILES:
0000    110 ;
0000    111         $BOOCMDDEF                      ; DEFINE SYSGEN COMMAND OPTIONS
0000    112         $CCBDEF                         ; DEFINE CHANNEL CONTROL BLOCK
0000    113         $DYNDEF                         ; DEFINE STRUCTURE TYPE CODES
0000    114         $IPLDEF                         ; IPL CONSTANTS
```

```
                        0000    115          $FABDEF                         ; RMS FAB DEFINITIONS
                        0000    116          $FIBDEF                         ; FILE INFORMATION BLOCK
                        0000    117          $PFLDEF                         ; PAGE FILE CONTROL BLOCK
                        0000    118          $PRMDEF                         ; DEFINE PARAMETER BLOCK
                        0000    119          $PTEDEF                         ; DEFINE PTE FIELDS AND CONSTANTS
                        0000    120          $RABDEF                         ; RMS RAB DEFINITIONS
                        0000    121          $SHRDEF                         ; SHARED MESSAGE DEFINITIONS
                        0000    122          $STSDEF                         ; STATUS MESSAGE DEFINITIONS
                        0000    123          $SYSGMSGDEF                     ; SYSGEN MESSAGE DEFINITIONS
                        0000    124          $TPADEF                         ; DEFINE TPARSE SYMBOLS
                        0000    125          $WCBDEF                         ; WINDOW CONTROL BLOCK
                        0000    126
                        0000    127 ;
                        0000    128 ; EQUATED SYMBOLS:
                        0000    129 ;
                00000100 0000   130          BUFFER_SIZE=256                 ; Buffer size for /OUTPUT= & /INPUT=
                00000080 0000   131          FILNAMSIZ=128
                        0000    132
                        0000    133 ;
                        0000    134 ; OWN STORAGE:
                        0000    135 ;
                        0000    136
                00000000 0000   137          .Psect PAGED_DATA       rd,wrt,noexe,quad
                        0000    138
             53 49 4C 2E 0000   139 DEFNAM:  .ASCII  /.LIS/                  ; INITIAL DEFAULT IS LISTING
                00000030 0004   140          .BLKB   48-4                    ; DEFAULT NAME GOES HERE
                000000B0 0030   141 EXPFNM:  .BLKB   FILNAMSIZ               ; EXPANDED NAME HERE
                00000130 00B0   142 RSLFNM:  .BLKB   FILNAMSIZ               ; RESULTANT NAME HERE
                000001B0 0130   143 OUTNAM:  .BLKB   FILNAMSIZ               ; OUTPUT FILE NAME HERE
                000001C4 01B0   144 RHBUF:   .BLKB   20                      ; RECORD HEADER BUFFER
                         01C4   145
      54 55 50 4E 49 24 53 59 53 01C4   146 INPNAM:  .ASCII  /SYS$INPUT/      ; INPUT NAME STRING
                00000009 01CD   147 INPNAMSZ=.-INPNAM                        ; SIZE OF INPUT STRING
                         01CD   148
   54 55 50 54 55 4F 24 53 59 53 01CD   149 OUTPNAM:.ASCII  /SYS$OUTPUT/     ; OUTPUT NAME STRING
                0000000A 01D7   150 OUTPNAMSZ=.-OUTPNAM                      ; SIZE OF OUTPUT STRING
                         01D7   151
52 41 50 2E 54 4C 55 41 46 45 44 01D7   152 PRMDEF:  .ASCII  /DEFAULT.PAR/    ; DEFAULT NAME STRING
                0000000B 01E2   153 PRMDEFSIZ=.-PRMDEF                       ; STRING LENGTH
                         01E2   154
2E 3A 4D 45 54 53 59 53 24 53 59 53 01E2   155 EXEDEF:  .ASCII  /SYS$SYSTEM:.EXE/ ; DEFAULT NAME FOR .EXE'S
                45 58 45 01EE
                0000000F 01F1   156 EXEDEFSIZ=.-EXEDEF                       ; STRING LENGTH
                         01F1   157
                      00 01F1   158 INP_OPEN: .BYTE 0                         ; INPUT OPEN FLAG
                      00 01F2   159 OUTP_OPEN:.BYTE 0                         ; OUTPUT OPEN FLAG
                         01F3   160
                         01F3   161
                00000213 01F3   162 ZEROES:  .BLKL   8                       ;
                         0213   163 ;
                         0213   164 ; RMS 32 DATA STRUCTURES
                         0213   165 ;
                         0213   166
                         0213   167          .ALIGN  QUAD                    ; ALIGN FOR PERFORMANCE
                         0218   168 RIO_OUTFAB::                            ; EXTERNAL NAME
                         0218   169          $FAB    FAC=PUT,-               ; OUTPUT FAB, ACCESS IS PUT
                         0218   170                  FNA=OUTPNAM,-           ; OUTPUT NAME STRING
```

```
0218   171                     FNS=OUTPNAMSZ,-          ; AND LENGTH
0218   172                     ORG=SEQ,-                ; SEQUENTIAL FILE
0218   173                     NAM=RIO_OUTNAM,-         ; ASSOCIATED NAME BLOCK
0218   174                     FOP=SUP,-                ; SUPERSEED IF NECESSARY
0218   175                     DNA=DEFNAM,-             ; DEFAULT NAME ADDRESS
0218   176                     RAT=CR                   ; DO CR/LF PER LINE
0268   177
0268   178 RIO_OUTNAM::                                ; OUTPUT NAME BLOCK
0268   179             $NAM    RSA=OUTNAM,-             ;
0268   180                     RSS=FILNAMSIZ,-          ;
0268   181                     ESA=OUTNAM,-             ; EXPANDED AND RESULT IN SAME PLACE
0268   182                     ESS=FILNAMSIZ,-          ;
0268   183
02C8   184 RIO_OUTRAB::                                ;
02C8   185             $RAB    RAC=SEQ,-                ; RECORD ACCESS IS SEQENTIAL
02C8   186                     RHB=RHBUF,-              ; RECORD HEADER BUFFER
02C8   187                     FAB=RIO_OUTFAB           ; ASSOCIATED FAB
030C   188
030C   189 RIO_INPFAB::                                ;
030C   190             $FAB    FAC=GET,-                ; FILE ACCESS IS GET(READ)
030C   191                     FNA=INPNAM,-             ; INPUT FILE NAME STRING ADDRESS
030C   192                     FNS=INPNAMSZ,-           ; INPUT STRING SIZE
030C   193                     NAM=RIO_INPNAM,-         ; ADDRESS OF ASSOCIATED NAME BLOCK
030C   194                     DNA=DEFNAM,-             ; ADDRESS OF DEFAULT NAME BUFFER
030C   195                     DNS=4                    ; DEFAULT SIZE STARTS AT 4
035C   196
035C   197 RIO_INPNAM::                                ;
035C   198             $NAM    ESA=EXPFNM,-             ; EXPANDED NAME BUFFER ADDRESS
035C   199                     ESS=FILNAMSIZ,-          ; MAXIMUM SIZE FOR EXPANDED NAME
035C   200                     RSA=RSLFNM,-             ; RESULTANT NAME
035C   201                     RSS=FILNAMSIZ,-          ; AND SIZE IS SAME
035C   202
03BC   203 RIO_INPRAB::                                ;
03BC   204             $RAB    RAC=SEQ,-                ; SEQUENTIAL ACCESS TO FILE
03BC   205                     ROP=<PMT,CVT>,-          ; PROMPT READ AND CONVERT TO UPPER CASE
03BC   206                     RHB=RHBUF,-              ; RECORD HEADER BUFFER
03BC   207                     FAB=RIO_INPFAB           ; ASSOCIATED FILE ACCESS BLOCK
0400   208
0400   209 ;
0400   210 ;       PARAMETER FILE INPUT/OUTPUT FAB/RAB
0400   211 ;
0400   212 PRM_INFAB:                                  ;
0400   213             $FAB    FAC=<GET,BIO>,-          ; READ ACCESS
0400   214                     DNA=PRMDEF,-             ; DEFAULT NAME
0400   215                     DNS=PRMDEFSIZ,-          ; DEFAULT NAME SIZE
0400   216                     NAM=RIO_INPNAM           ; NAME BLOCK
0450   217 PRM_INRAB:                                  ;
0450   218             $RAB    FAB=PRM_INFAB            ; ASSOCIATED FAB
0494   219 ;
0494   220 ;       PARAMETER FILE OUTPUT RAB/RAB
0494   221 ;
0494   222 PRM_OUTFAB:                                 ;
0494   223             $FAB    FAC=<PUT,TRN>,-          ;
0494   224                     DNA=PRMDEF,-             ; DEFAULT NAME
0494   225                     DNS=PRMDEFSIZ,-          ; DEFAULT NAME LENGTH
0494   226                     RFM=FIX,-                ; FIXED RECORD FORMAT
0494   227                     MRS=512,-                ;
```

```
                     0494   228                    FOP=CTG,-           ; CONTIGUOUS
                     0494   229                    ALQ=15,-            ; Initially allocate 15 blocks
                     0494   230                    NAM=RIO_INPNAM      ; NAME BLOCK
                     04E4   231 PRM_OUTRAB:                            ; OUTPUT RAB
                     04E4   232             $RAB    FAB=PRM_OUTFAB,-            ; ASSOCIATED FAB
                     04E4   233                     RSZ=512,-                   ; RECORD SIZE
                     04E4   234                     RBF=BOO$AB_PRMBUF           ; BUFFER ADDRESS
                     0528   235
                     0528   236 ;
                     0528   237 ;       SWAPFILE FAB  USED TO OBTAIN STARTING LBN OF SWAPFILE
                     0528   238 ;
                     0528   239 SWAPFHCFAB:      $FAB    FAC=<GET,PUT,BIO>,- ; FILE ACCESS IS GET & PUT (READ & WRITE
                     0528   240                    DNA=SWPDEFNAM,- ; DEFAULT FILE NAME STRING ADDRESS
                     0528   241                    DNS=SWPDEFNAMSZ,-       ; DEFAULT FILE NAME STRING SIZE
                     0528   242                    RFM=FIX,-           ; FIXED RECORD FORMAT
                     0528   243                    MRS=512,-           ; MAXIMUM RECORD SIZE OF ONE PAGE
                     0528   244                    NAM=RIO_INPNAM,-    ; NAME BLOCK ADDRESS
                     0528   245                    RTV=255,-           ; LET ACP COMPUTE LARGEST RETRIVAL WINDOW
                     0528   246                    XAB=SWAPFHCXAB      ; EXTENDED ATTRIBUTE BLOCK
                     0578   247 SWAPFHCXAB:      $XABFHC             ; EXTENDED ATTRIBUTE BLOCK FOR FILE HEADER
                     05A4   248 SWAPALLXAB:      $XABALL -           ; EXTENDED ATTRIBUTE BLOCK FOR ALLOCATION
                     05A4   249                    VOL=1,-             ; ALWAYS ALLOCATE ON RELATIVE VOLUME #1
                     05A4   250                    AOP=<CTG,HRD>,-     ; CONTIGUOUS AND FORCE VOLUME PLACEMENT
                     05A4   251                    ALN=LBN             ; (MUST SPECIFY SO VOL IS USED)
                     05C4   252 SWAPALLXABNC:    $XABALL -           ; EXTENDED ATTRIBUTE BLOCK FOR NON-CONTIGUOUS A_OCA
                     05C4   253                    AOP=<CBT>           ; CONTIGUOUS BEST TRY
                     05E4   254
                     05E4   255 SWAP_RAB:                              ; OUTPUT RAB
                     05E4   256             $RAB    FAB=SWAPFHCFAB,-            ; ASSOCIATED FAB
                     05E4   257                     RAC=KEY,-
                     05E4   258                     RSZ=512,-                   ; RECORD SIZE
                     05E4   259                     RBF=ZEROES                  ; BUFFER ADDRESS
                     0628   260
          53 59 53 2E 0628   261 SWPDEFNAM:      .ASCII  /.SYS/              ; DEFAULT NAME STRING
             00000004  062C   262 SWPDEFNAMSZ=.-SWPDEFNAM          ; SIZE OF DEFAULT NAME STRING
                     062C   263
                     062C   264 ;
                     062C   265 ;       PARSED PARAMETERS FOR SWAP FILE CREATION AND INSTALLATION
                     062C   266 ;
                     062C   267 BOO$GL_SIZE:                       ; FILE SIZE
             00000000 062C   268             .LONG   0                   ;
                     0630   269 BOO$GL_INDEX:                      ; RETURNED PFL INDEX
             00000000 0630   270             .LONG   0
                     0634   271
                     0634   272 ;
                     0634   273 ; OWN STORAGE:
                     0634   274 ;
                     0634   275
                     0634   276
3A 54 55 50 54 55 4F 0634   277 OUTFNM: .ASCII  \OUTPUT:\
                     063B   278
             00000007 063B   279             OUTSIZ=.-OUTFNM
                     063B   280
```

```
                     063B   282          .SBTTL   RIO$INPUT - RMS INPUT INTERFACE
                     063B   283 ;++
                     063B   284 ; FUNCTIONAL DESCRIPTION:
                     063B   285 ;       RIO$INPUT READS A RECORD FROM THE DEVICE OR FILE DESCRIBED BY THE
                     063B   286 ;       LOGICAL NAME 'INPUT' INTO THE BUFFER DESCRIBED BY THE ARGUMENTS BUF
                     063B   287 ;       AND SIZE.  THE ACTUAL SIZE READ IS RETURNED IN R1.
                     063B   288 ;
                     063B   289 ; CALLING SEQUENCE:
                     063B   290 ;       CALLG   ARGLIST,RIO$INPUT
                     063B   291 ;
                     063B   292 ; INPUT PARAMETERS:
                     063B   293 ;       BUF(AP) -       BUFFER ADDRESS
                     063B   294 ;       SIZE(AP)-       MAXIMUM SIZE OF READ
                     063B   295 ;
                     063B   296 ; OUTPUT PARAMETERS:
                     063B   297 ;       R0      -       COMPLETION STATUS
                     063B   298 ;       R1      -       ACTUAL SIZE OF RECORD IN BYTES
                     063B   299 ;
                     063B   300 ; COMPLETION STATUS CODES:
                     063B   301 ;       STANDARD RMS COMPLETION STATUS CODES
                     063B   302 ;
                     063B   303 ;--
           00000008  063B   304 SIZE=8                                   ; OFFSET TO SIZE ARGUMENT
                     063B   305
                00000000   306          .Psect PAGED_CODE       rd,nowrt,exe,long
                     0000   307
                     0000   308 RIO$INPUT::                              ; INPUT INTERFACE ROUTINE
                   01FC 0000   309          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8>; REGISTER SAVE MASK
     52    000003BC'EF   9E 0002   310          MOVAB   RIO_INPRAB,R2          ; GET INPUT RAB ADDRESS
       26  000001F1'EF   E8 0009   311          BLBS    INP_OPEN,15$           ; BR IF INPUT IS ALREADY OPEN
                  00   DD 0010   312          PUSHL   #0                     ; OUTPUT OPEN FAILURE AST
          0000030C'EF   9F 0012   313          PUSHAB  RIO_INPFAB             ; THE FAB
    00000000'EF   02   FB 0018   314          CALLS   #2,SYS$OPEN            ; OPEN THE FILE
                01 50   E8 001F   315          BLBS    R0,10$                 ; BR IF OPEN OK
                        04 0022   316          RET                            ; RETURN ERROR STATUS
                           0023   317 10$:                                   ;
                           0023   318          $CONNECT (R2)                  ; CONNECT TO INPUT STREAM
                23 50   E9 002C   319          BLBC    R0,30$                 ; BR IF ERROR
    000001F1'EF   01   90 002F   320          MOVB    #1,INP_OPEN            ; MARK INPUT OPEN AND CONNECTED
                           0036   321
       20 A2    08 AC   B0 0036   322 15$:     MOVW    SIZE(AP),RAB$W_USZ(R2) ; SET SIZE FOR INPUT
     24 A2   000C'CC   D0 003B   323          MOVL    BUF(AP),RAB$L_UBF(R2)  ; AND BUFFER ADDRESS
                           0041   324          $GET    (R2)                   ; GET A RECORD
                01 50   E8 004A   325          BLBS    R0,20$                 ; SUCCESS
                        04 004D   326          RET                            ; RETURN ERROR STATUS
          51    22 A2   3C 004E   327 20$:     MOVZWL  RAB$W_RSZ(R2),R1       ; GET SIZE OF RECORD
                        04 0052   328 30$:     RET                            ; RETURN
                           0053   329
```

```
                     0053   331          .SBTTL  RIO$OUTPUT - RMS OUTPUT INTERFACE ROUTINE
                     0053   332 ;++
                     0053   333 ; FUNCTIONAL DESCRIPTION:
                     0053   334 ;       RIO$OUTPUT WRITES THE BUFFER DESCRIBED BY THE BUF AND SIZE
                     0053   335 ;       ARGUMENTS TO THE DEVICE OR FILE SPECIFIED BY THE LOGICAL NAME
                     0053   336 ;       OUTPUT.
                     0053   337 ;
                     0053   338 ;       RIO$OUTPUTC WRITES THE COUNTED ASCII STRING SPECIFIED BY THE
                     0053   339 ;       BUF ARGUMENT TO THE DEVICE OR FILE SPECIFIED BY THE LOGICAL NAME
                     0053   340 ;       OUTPUT.
                     0053   341 ;
                     0053   342 ; CALLING SEQUENCE:
                     0053   343 ;       CALLG   ARGLIST,RIO$OUTPUT/RIO$OUTPUTC
                     0053   344 ;
                     0053   345 ; INPUT PARAMETERS:
                     0053   346 ;       BUF(AP) -       ADDRESS OF BUFFER OR ADDRESS OF COUNTED STRING IF
                     0053   347 ;                       RIO$OUTPUTC.
                     0053   348 ;       SIZE(AP)-       NUMBER OF BYTES TO BE OUTPUT (RIO$OUTPUT ONLY)
                     0053   349 ;
                     0053   350 ;
                     0053   351 ; IMPLICIT INPUTS:
                     0053   352 ;       INPUT RAB AND FAB
                     0053   353 ;
                     0053   354 ; OUTPUT PARAMETERS:
                     0053   355 ;       R0      -       COMPLETION STATUS CODE FROM RMS
                     0053   356 ;
                     0053   357 ; COMPLETION CODES:
                     0053   358 ;       STANDARD RMS COMPLETION STATUS CODES
                     0053   359 ;
                     0053   360 ;--
                     0053   361
00000004             0053   362 BUF=4                                   ; OFFSET TO BUFFER ADDRESS ARGUMENT
                     0053   363
                     0053   364 RIO$OUTPUTC::                          ; OUTPUT COUNTED STRING
                0000 0053   365          .WORD   0                     ; ENTRY MASK
     51   04 AC     D0 0055   366          MOVL    BUF(AP),R1            ; GET BUFFER ADDRESS
        7E   81     9A 0059   367          MOVZBL  (R1)+,-(SP)           ; PUSH SIZE OF WRITE
             51     DD 005C   368          PUSHL   R1                    ; AND BUFFER ADDRESS
     63'AF   02     FB 005E   369          CALLS   #2,B^RIO$OUTPUT       ; CALL OUTPUT
             04     0062   370          RET                           ; AND RETURN
                    0063   371
                    0063   372 RIO$OUTPUT::                           ; RMS OUTPUT INTERFACE ROUTINE
                00FC 0063   373          .WORD   ^M<R2,R3,R4,R5,R6,R7> ; REGISTER SAVE MASK
52 000002C8'EF      9E 0065   374          MOVAB   RIO_OUTRAB,R2         ; GET ADDRESS OF OUTPUT RAB
 48 000001F2'EF     E8 006C   375          BLBS    OUTP_OPEN,10$         ; BR IF FILE ALREADY OPEN
    00000273'EF     94 0073   376          CLRB    NAM$B_ESL+RIO_OUTNAM  ; ZERO EXPANDED NAME LENGTH
                    0079   377          $CREATE RIO_OUTFAB            ; CREATE TYPING FILE
           45 50    E9 0086   378          BLBC    R0,20$                ; BR IF OPEN FAILED
                    0089   379
    00010000 8F     CA 0089   380          BICL    #BOOCMD$M_TERMINAL,-
    00000000'EF             008F   381                  BOO$GL_CMDOPT         ; ASSUME NOT A TERMINAL DEVICE
    00000000'8F     E1 0094   382          BBC     #DEV$V_TRM,-
    00000258'EF             009A   383                  RIO_OUTFAB+FAB$L_DEV,-
              0B            009F   384                  5$                    ; BRANCH IF NOT
    00010000 8F     C8 00A0   385          BISL    #BOOCMD$M_TERMINAL,-
    00000000'EF             00A6   386                  BOO$GL_CMDOPT         ; SET AS A TERMINAL DEVICE
                    00AB   387
```

```
                        00AB    388 5$:     $CONNECT  (R2)                          ; CONNECT TO OUTPUT STREAM
000001F2'EF    01   90  00B4    389          MOVB    #1,OUTP_OPEN                    ; MARK OUTPUT STREAM OPEN
   28 A2    04 AC   D0  00BB    390 10$:     MOVL    BUF(AP),RAB$L_RBF(R2)           ; SET OUTPUT BUFFER
   22 A2    08 AC   B0  00C0    391          MOVW    SIZE(AP),RAB$W_RSZ(R2)          ; AND SIZE
                        00C5    392          $PUT    (R2)                            ; PRINT LINE
                    04  00CE    393 20$:     RET                                     ; RETURN
                        00CF    394
                        00CF    395
```

```
                                00CF   397          .SBTTL  BOO$READPROMPT - Prompt and read input string
                                00CF   398 ;++
                                00CF   399 ; Functional Description:
                                00CF   400 ;       BOO$READPROMPT outputs the specified ASCIZ prompt string on the
                                00CF   401 ;       console terminal then checks the count of characters to be read.
                                00CF   402 ;       If zero it exits, otherwise it reads the console terminal until
                                00CF   403 ;       either a carriage return is encountered or the character count
                                00CF   404 ;       is satisfied.  The specified buffer is filled with an ASCIC
                                00CF   405 ;       string containing the characters read but not including the
                                00CF   406 ;       terminating carriage return.
                                00CF   407 ;
                                00CF   408 ; Calling Sequence:
                                00CF   409 ;       CALLG   ARGLIST,BOO$READPROMPT
                                00CF   410 ;
                                00CF   411 ; Input Parameters:
                                00CF   412 ;       PROMPT(AP)   -  Address of ASCIZ prompt string
00000004                        00CF   413          PROMPT  =  4
                                00CF   414 ;
                                00CF   415 ;       SIZE(AP)     -  Maximum length of input string
00000008                        00CF   416          SIZE    =  8
                                00CF   417 ;                        Note: if size is zero, then nothing is read
                                00CF   418 ;                              and only the prompt string is written.
                                00CF   419 ;
                                00CF   420 ;       BUF(AP)      -  Address of input buffer
0000000C                        00CF   421          BUF     =  12
                                00CF   422 ;
                                00CF   423 ; Output Parameters:
                                00CF   424 ;       R0 - Completion status code (always SS$_NORMAL)
                                00CF   425 ;
                                00CF   426 ;       Buffer located by BUF(AP) will be filled with the string
                                00CF   427 ;       read as an ASCIC string.
                                00CF   428 ;
                                00CF   429 ;--
                                00CF   430 BOO$READPROMPT::                             ;
                         00FC   00CF   431          .WORD   ^M<R2,R3,R4,R5,R6,R7>       ;
04 BC    FA00 8F    00    3A   00D1   432          LOCC    #0,#64000,@PROMPT(AP)       ; FIND END OF PROMPT STRING
     57    51    04 AC    C3   00D8   433          SUBL3   PROMPT(AP),R1,R7            ; COMPUTE CHARACTER COUNT
     52    000003BC'EF    9E   00DD   434          MOVAB   RIO_INPRAB,R2               ; GET INPUT RAB ADDRESS
       26 000001F1'EF    E8   00E4   435          BLBS    INP_OPEN,15$                ; BR IF INPUT IS ALREADY OPEN
                  00     DD   00EB   436          PUSHL   #0                          ; OUTPUT OPEN FAILURE AST
          0000030C'EF    9F   00ED   437          PUSHAB  RIO_INPFAB                  ; THE FAB
     000000C0'EF    02    FB   00F3   438          CALLS   #2,SYS$OPEN                 ; OPEN THE FILE
                 01 50    E8   00FA   439          BLBS    R0,10$                      ; BR IF OPEN OK
                          04   00FD   440          RET                                 ; RETURN ERROR STATUS
                               00FE   441 10$:                                        ;
                               00FE   442          $CONNECT (R2)                       ; CONNECT TO INPUT STREAM
                 42 50    E9   0107   443          BLBC    R0,30$                      ; BR IF ERROR
     000001F1'EF    01    90   010A   444          MOVB    #1,INP_OPEN                 ; MARK INPUT OPEN AND CONNECTED
        20 A2    08 AC    B0   0111   445 15$:     MOVW    SIZE(AP),RAB$W_USZ(R2)      ; SET SIZE FOR INPUT
                    53    13   0116   446          BEQL    NOREAD                      ; IF NO BUFFER THEN NO READ
           34 A2    57    90   0118   447          MOVB    R7,RAB$B_PSZ(R2)            ; SET SIZE FOR PROMPT
        30 A2    04 AC    D0   011C   448          MOVL    PROMPT(AP),RAB$L_PBF(R2)    ; AND PROMPT ADDRESS
  24 A2    0C AC    01    C1   0121   449          ADDL3   #1,BUF(AP),RAB$L_UBF(R2)    ; AND BUFFER ADDRESS
                 0C BC    94   0127   450          CLRB    @BUF(AP)                    ; INITIALIZE COUNT FOR READ STRING
     50    00000000'EF    7D   012A   451          MOVQ    BOO$GQ_CMDESC,R0            ; GET COMMAND STRING DESCRIPTOR
                    50    D5   0131   452          TSTL    R0                          ; IS THERE ANY COMMAND STRING
                    2E    19   0133   453          BLSS    50$                         ; NO, ALREADY USED IT
```

```
             16    14  0135              BGTR     40$                        ; YES, USE IT AS RECORD
                       0137              $GET     (R2)                       ; GET A RECORD
          01 50    E8  0140              BLBS     R0,20$                     ; SUCCESS
                   04  0143              RET                                 ; RETURN ERROR STATUS
     51    22 A2   3C  0144      20$:    MOVZWL   RAB$W_RSZ(R2),R1           ; GET SIZE OF RECORD
     0C BC    51   33  0148              CVTWB    R1,@BUF(AP)                ; RETURN SIZE OF RECORD
                   04  014C      30$:    RET                                 ; RETURN
     53    0C AC   D0  014D      40$:    MOVL     BUF(AP),R3                 ; GET BUFFER ADDRESS
        83    50   33  0151              CVTWB    R0,(R3)+                   ; SET COUNT FOR BUFFER
   63   61    50   28  0154              MOVC3    R0,(R1),(R3)               ; COPY COMMAND STRING
        50    01   D0  0158              MOVL     #1,R0                      ; SET SUCCESS STATUS
00000000'EF   01   CE  015B              MNEGL    #1,BOO$GQ_CMDESC           ; MARK AS USED
                   04  0162              RET                                 ;
50    00000000'8F  D0  0163      50$:    MOVL     #RMS$_EOF,R0               ; GIVE END OF FILE STATUS
                   04  016A              RET                                 ; AND RETURN
                       016B      NOREAD:                                     ;
52    000002C8'EF  9E  016B              MOVAB    RIO_OUTRAB,R2              ; GET ADDRESS OF OUTPUT RAB
  48  000001F2'EF  E8  0172              BLBS     OUTP_OPEN,10$              ; BR IF FILE ALREADY OPEN
      00000273'EF  94  0179              CLRB     NAM$B_ESL+RIO_OUTNAM       ; ZERO EXPANDED NAME LENGTH
                       017F              $CREATE  RIO_OUTFAB                 ; CREATE TYPING FILE
          44 50    E9  018C              BLBC     R0,20$                     ; BR IF OPEN FAILED
                       018F
      00010000 8F  CA  018F              BICL     #BOOCMD$M_TERMINAL,-
      00000000'EF      0195                       BOO$GL_CMDOPT              ; ASSUME NOT A TERMINAL DEVICE
      00000000'8F  E1  019A              BBC      #DEV$V_TRM,-
      00000258'EF      01A0                       RIO_OUTFAB+FAB$L_DEV,-
               0B      01A5                       5$                         ; BRANCH IF NOT
      00010000 8F  C8  01A6              BISL     #BOOCMD$M_TERMINAL,-
      00000000'EF      01AC                       BOO$GL_CMDOPT              ; SET AS A TERMINAL DEVICE
                       01B1
                       01B1      5$:     $CONNECT  (R2)                      ; CONNECT TO OUTPUT STREAM
000001F2'EF   01   90  01BA              MOVB     #1,OUTP_OPEN               ; MARK OUTPUT STREAM OPEN
  28 A2    04 AC   D0  01C1      10$:    MOVL     PROMPT(AP),RAB$L_RBF(R2)   ; SET OUTPUT BUFFER
     22 A2    57   B0  01C6              MOVW     R7,RAB$W_RSZ(R2)           ; AND SIZE
                       01CA              $PUT     (R2)                       ; PRINT LINE
                   04  01D3      20$:    RET                                 ; RETURN
```

```
                                01D4    491          .SBTTL  BOO$FILOPEN - Routine to open input file
                                01D4    492 ;
                                01D4    493 ; Input Parameters:
                                01D4    494 ;       R7 - Address of file name descriptor
                                01D4    495 ; Output Parameters:
                                01D4    496 ;       R0 - Completion status
                                01D4    497 ;       R8 - RAB address
                                01D4    498 ;
                                01D4    499
                                01D4    500 BOO$FILOPENW::                          ; Open file for reads and writes
          58    00000400'EF     9E      01D4    501          MOVAB   PRM_INFAB,R8       ; Get address of FAB
                  16 A8   01    88      01DB    502          BISB    #FAB$M_PUT,FAB$B_FAC(R8); Set PUT bit
                          1D    11      01DF    503          BRB     FILOPEN
                                01E1    504
                                01E1    505 BOO$EXEOPEN::                           ; Open with default SYS$SYSTEM:.EXE
00000430'EF     000001E2'EF     9E      01E1    506          MOVAB   EXEDEF,PRM_INFAB+FAB$L_DNA
          00000435'EF   0F      90      01EC    507          MOVB    #EXEDEFSIZ,PRM_INFAB+FAB$B_DNS
                                01F3    508
                                01F3    509 BOO$FILOPEN::                           ; Open file for read only
          58    00000400'EF     9E      01F3    510          MOVAB   PRM_INFAB,R8       ; Get address of FAB
                  16 A8   01    8A      01FA    511          BICB    #FAB$M_PUT,FAB$B_FAC(R8); Clear PUT bit
                                01FE    512
                                01FE    513 FILOPEN:
                  34 A8   67    90      01FE    514          MOVB    (R7),FAB$B_FNS(R8)     ; Set size of file name
               2C A8   04 A7    D0      0202    515          MOVL    4(R7),FAB$L_FNA(R8)    ; and address of name string
                                0207    516          $CLOSE  FAB=(R8)               ; Close if open
                                0210    517          $OPEN   FAB=(R8)               ; Open file for read
       30 A8    000001D7'EF     9E      0219    518          MOVAB   PRMDEF,FAB$L_DNA(R8)   ; Go back to ususal default
                  35 A8   0B    90      0221    519          MOVB    #PPMDEFSIZ,FAB$B_DNS(R8)
                        07 50   E8      0225    520          BLBS    R0,10$                 ; Continue if successful
                           FDD5'30      0228    521          BSBW    PUTERROR               ; Output error message
                       50   02  CE      022B    522          MNEGL   #2,R0                  ; Return failure
                                05      022E    523          RSB                            ;
00000000'EF     000000B0'EF     DE      022F    524 10$:     MOVAL   RSLFNM,BOO$GL_FILEADDR ; Set expanded file address
                0000035F'EF     90      023A    525          MOVB    RIO_INPNAM+NAM$B_RSI   ;
                00000000'EF             0240    526                  BOO$GB_FILELEN         ; Set expanded file length
          58    00000450'EF     9E      0245    527          MOVAB   PRM_INRAB,R8       ; Get RAB address
                                024C    528          $CONNECT RAB=(R8)              ; Connect record stream
                        58  01  D0      0255    529          MOVL    #1,R8                  ; Return starting virtual block
                                05      0258    530          RSB                            ;
                                0259    531
```

```
                     0259    533          .SBTTL  BOO$UFOOPEN - Routine to open input file for user access
                     0259    534 ;
                     0259    535 ; Input Parameters:
                     0259    536 ;       R7 - Address of file name descriptor
                     0259    537 ; Output Parameters:
                     0259    538 ;       R0 - Completion status
                     0259    539 ;       R1 - Channel
                     0259    540 ;
                     0259    541
                     0259    542 BOO$UFOOPEN::                                   ; Open file for read only
   58   00000400'EF   9E   0259    543          MOVAB   PRM_INFAB,R8               ; Get address of FAB
30 A8   000001E2'EF   9E   0260    544          MOVAB   EXEDEF,FAB$L_DNA(R8)       ; Set default name string to
     35 A8    0F      90   0268    545          MOVB    #EXEDEFSIZ,FAB$B_DNS(R8)   ; SYS$SYSTEM:.EXE
     16 A8    01      8A   026C    546          BICB    #FAB$M_PUT,FAB$B_FAC(R8)   ; Clear PUT bit
04 A8   00020000 8F      C8   0270    547          BISL    #FAB$M_UFO,FAB$L_FOP(R8)   ; UFO
     34 A8    67      90   0278    548          MOVB    (R7),FAB$B_FNS(R8)         ; Set size of file name
   2C A8    04 A7      D0   027C    549          MOVL    4(R7),FAB$L_FNA(R8)        ; and address of name string
                         0281    550          $CLOSE  FAB=(R8)                   ; Close if open
                         028A    551          $OPEN   FAB=(R8)                   ; Open file for read
30 A8   000001D7'EF   9E   0293    552          MOVAB   PRMDEF,FAB$L_DNA(R8)       ; Go back to usual default
     35 A8    0B      90   029B    553          MOVB    #PRMDEFSIZ,FAB$B_DNS(R8)
        07 50         E8   029F    554          BLBS    R0,10$                     ; Continue if successful
          FD5B'       30   02A2    555          BSBW    PUTERROR                   ; Output error message
     50   02          CE   02A5    556          MNEGL   #2,R0                      ; Return failure
                      05   02A8    557          RSB                                ;
04 A8   00020000 8F      CA   02A9    558 10$:     BICL    #FAB$M_UFO,FAB$L_FOP(R8)   ; Next caller doesn't want UFO
     51   0C A8      3C   02B1    559          MOVZWL  FAB$L_STV(R8),R1           ; Channel
                      05   02B5    560          RSB
                         02B6    561
```

```
                     02B6    563          .SBTTL  BOO$FILCLOSE - Routine to close currently open file
                     02B6    564 ;
                     02B6    565 ; Input Parameters:
                     02B6    566 ;       None
                     02B6    567 ; Output Parameters:
                     02B6    568 ;       R0 - Completion status
                     02B6    569 ;       R8 - FAB address
                     02B6    570 ;
                     02B6    571
                     02B6    572 BOO$FILCLOSE::                              ; Open file for reads and writes
  58 00000400'EF   9E 02B6    573          MOVAB   PRM_INFAB,R8               ; Get address of FAB
                     02BD    574          $CLOSE  FAB=(R8)                   ; Close if open
                  05 02C6    575          RSB
                     02C7    576
```

```
                     02C7   578          .SBTTL  BOO$READFILE - Read parameter file
                     02C7   579 ;
                     02C7   580 ; Input Parameters:
                     02C7   581 ;       R6 - Buffer address
                     02C7   582 ;       R8 - VBN
                     02C7   583 ;       R9 - Count of pages to read
                     02C7   584 ;
                     02C7   585 ; Output Parameters:
                     02C7   586 ;       R0 - Completion status
                     02C7   587 ;
                     02C7   588 BOO$READFILE::                               ;
51   00000450'EF  9E 02C7   589          MOVAB   PRM_INRAB,R1               ; Get address of input RAB
 20 A1    0200 8F  B0 02CE   590          MOVW    #512,RAB$W_USZ(R1)         ; Set record size
      24 A1    56  D0 02D4   591          MOVL    R6,RAB$L_UBF(R1)           ; Set buffer address
      38 A1    58  D0 02D8   592          MOVL    R8,RAB$L_BKT(R1)           ; Set VBN to read
                   02DC   593          $READ   RAB=(R1)                   ;
          11 50    E8 02E5   594          BLBS    R0,10$                     ;
00000000'8F   50    D1 02E8   595          CMPL    R0,#RMS$_EOF               ; Is it end of file?
              12    13 02EF   596          BEQL    20$                        ; Branch if yes
50   007C8102 8F    D0 02F1   597          MOVL    #SYSG$_INVPARFIL,R0        ; Invalid parameter file
                     05 02F8   598          RSB                                ;  and return
    56   0200 C6  9E 02F9   599 10$:    MOVAB   512(R6),R6                 ; Advance buffer pointer
              58    D6 02FE   600          INCL    R8                         ; and block number
            C4 59   F5 0300   601          SOBGTR  R9,BOO$READFILE            ; Read all required blocks
          50   01   3C 0303   602 20$:    MOVZWL  #1,R0                      ; Return success
                     05 0306   603          RSB                                ;
```

```
                          0307   605          .SBTTL  BOO$WRITEFILE - Write blocks to file
                          0307   606 ;
                          0307   607 ; Input Parameters:
                          0307   608 ;       R6 - Buffer address
                          0307   609 ;       R8 - VBN
                          0307   610 ;       R9 - Count of pages to write
                          0307   611 ;
                          0307   612 ; Output Parameters:
                          0307   613 ;       R0 - Completion status
                          0307   614 ;
                          0307   615 BOO$WRITEFILE::
51   00000450'EF    9E    0307   616          MOVAB   PRM_INRAB,R1            ; Use input RAB for output too
 22 A1    0200 8F    B0    030E   617          MOVW    #512,RAB$W_RSZ(R1)      ; Set block size
      28 A1    56    D0    0314   618          MOVL    R6,RAB$L_RBF(R1)        ; Set buffer address
      38 A1    58    D0    0318   619          MOVL    R8,RAB$L_BKT(R1)        ; Set VBN to write
                       031C   620          $WRITE  RAB=(R1)                ; Do it!
            21 50    E9    0325   621          BLBC    R0,20$                  ; Error
    56   0200 C6    9E    0328   622          MOVAB   512(R6),R6              ; Advance buffer pointer
               58    D6    032D   623          INCL    R8                      ; and block number
            D5 59    F5    032F   624          SOBGTR  R9,BOO$WRITEFILE        ; Repeat
                       0332   625
51   00000400'EF    9E    0332   626          MOVAB   PRM_INFAB,R1            ; Get address of FAB
                       0339   627          $CLOSE  FAB = (R1)              ; Close file
            04 50    E9    0342   628          BLBC    R0,20$                  ; Error
         50   01    3C    0345   629          MOVZWL  #1,R0                   ; Return success
                       05    0348   630          RSB
                       0349   631
            FCB4'    30    0349   632 20$:     BSBW    PUTERROR                ; Output error message
         50   02    CE    034C   633          MNEGL   #2,R0                   ; Set error status
                       05    034F   634          RSB
```

```
                                0350   636          .SBTTL  BOO$WRTFILE - Routine to write parameter file
                                0350   637 ;
                                0350   638 ; Input parameters:
                                0350   639 ;       TPA$L_TOKENCNT(AP) - SIZE OF FILE NAME
                                0350   640 ;       TPA$L_TOKENPTR(AP) - ADDRESS OF FILE NAME
                                0350   641 ;
                                0350   642 ; Output Parameters:
                                0350   643 ;       R0 - Completion status code
                                0350   644 ;
                                0350   645 BOO$WRTFILE::                                    ;
                          03FC  0350   646          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9>     ;
      57   00000494'EF    9E  0352   647          MOVAB   PRM_OUTFAB,R7                   ; Get FAB address
         2C A7    14 AC    D0  0359   648          MOVL    TPA$L_TOKENPTR(AP),FAB$L_FNA(R7); Set into FAB
         34 A7    10 AC    90  035E   649          MOVB    TPA$L_TOKENCNT(AP),FAB$B_FNS(R7); and set size of name
      59   00000000'EF    9E  0363   650          MOVAB   BOO$A_PRMBLK,R9                 ; Get base of parameter blocks
      58   00000000'EF    9E  036A   651          MOVAB   BOO$AB_PRMBUF,R8                ; And set base of buffer
                                0371   652 ;
                                0371   653 ; Set startup filename
                                0371   654 ;
   68 00000000'EF    20    28  0371   655          MOVC3   #32,EXE$GT_STARTUP,(R8) ; Copy name of startup file
                  50    58    D0  0379   656          MOVL    R8,R0                   ; Save base of buffer
                  58    20    C0  037C   657          ADDL    #32,R8                  ; Advance buffer pointer
                                037F   658
                                037F   659 ;
                                037F   660 ; Loop through all parameters
                                037F   661 ;
                                037F   662
                     69    D5  037F   663 5$:      TSTL    (R9)                    ; Check for end of buffer
                     4D    13  0381   664          BEQL    8$                      ; Yes, write file
            88    16 A9    7D  0383   665          MOVQ    PRM$T_NAME(R9),(R8)+    ; Copy name
            88    1E A9    7D  0387   666          MOVQ    PRM$T_NAME+8(R9),(R8)+  ; string to buffer
         1F 10 A9    10    E1  038B   667          BBC     #PRM$V_ASCII,PRM$L_FLAGS(R9),6$; Branch if not an ascii parameter
            51    14 A9    9A  0390   668          MOVZBL  PRM$B_SIZE(R9),R1       ; Get size in bits
      51    51    FD 8F    78  0394   669          ASHL    #-3,R1,R1               ; Get size in bytes
         7E    51    03    C1  0399   670          ADDL3   #3,R1,-(SP)             ; Round size up to next longword boundary
               6E    03    CA  039D   671          BICL2   #3,(SP)                 ;
      20    00 B9    51    2C  03A0   672          MOVC5   R1,@PRM$L_ADDR(R9),#^A/ /,-; Put value in buffer
               68    6E        03A5   673                  (SP),(R8)               ;
               58    8E    C0  03A7   674          ADDL2   (SP)+,R8                ; Advance buffer pointer
               59    32    C0  03AA   675          ADDL    #PRM$C_LENGTH,R9        ; Next parameter
                     D0    11  03AD   676          BRB     5$                      ;
            51    15 A9    9A  03AF   677 6$:      MOVZBL  PRM$B_POS(R9),R1        ; Get position
            14 A9    51    EE  03B3   678          EXTV    R1,PRM$B_SIZE(R9),-
            52    00 B9        03B7   679                  @PRM$L_ADDR(R9),R2      ; Extract value
                     0C    E1  03BA   680          BBC     #PRM$V_NEG,-
               03 10 A9        03BC   681                  PRM$L_FLAGS(R9),7$      ; Branch if no negation needed
               52    52    CE  03BF   682          MNEGL   R2,R2                   ; Convert to negative
   52  52    14 A9    00    EF  03C2   683 7$:      EXTZV   #0,PRM$B_SIZE(R9),R2,R2 ; Convert value to unsigned form
               88    52    D0  03C8   684          MOVL    R2,(R8)+                ; Put value in buffer
               59    32    C0  03CB   685          ADDL    #PRM$C_LENGTH,R9        ; Next parameter
                     AF    11  03CE   686          BRB     5$                      ;
                                03D0   687
                     88    D4  03D0   688 8$:      CLRL    (R8)+                   ; Mark end of buffer
               58    50    C2  03D2   689          SUBL    R0,R8                   ; Size of buffer
         58    01FF C8    9E  03D5   690          MOVAB   511(R8),R8              ; Round to page bound
      58    58    F7 8F    78  03DA   691          ASHL    #-9,R8,R8               ; Compute file size in pages
            10 A7    58    D0  03DF   692          MOVL    R8,FAB$L_ALQ(R7)        ; Set size of file to create
```

```
                        03E3   693          $CREATE FAB=(R7)                ; Create and open file
                     08 50  E8  03EC   694          BLBS    R0,10$          ; Continue if success
          50  007C80FA 8F     DO  03EF   695          MOVL    #SYSG$_CREPARFIL,R0     ; Set message
                        04  03F6   696          RET                             ;
                        03F7   697 10$:       $CONNECT RAB=PRM_OUTRAB         ; Connect output RAB
          55  000004E4'EF     9E  0404   698          MOVAB   PRM_OUTRAB,R5           ; Get base address of output RAB
    28 A5     00000000'EF     9E  040B   699          MOVAB   BOO$AB_PRMBUF,RAB$L_RBF(R5)     ; Set base of buffer
                        0413   700 20$:       $PUT    RAB=(R5)                ; Write one block
    28 A5     00000200 8F     C0  041C   701          ADDL    #512,RAB$L_RBF(R5)      ; Advance buffer pointer
                     EC 58  F5  0424   702          SOBGTR  R8,20$                  ; Do them all
                        0427   703          $CLOSE  FAB=(R7)                ;
                        04  0430   704          RET                             ;
```

```
                                0431    706          .SBTTL  BOO$WRTSYSPARFILE - Routine to write the system parameter file
                                0431    707 ;
                                0431    708 ; Input parameters:
                                0431    709 ;       TPA$L_TOKENCNT(AP) - SIZE OF FILE NAME
                                0431    710 ;       TPA$L_TOKENPTR(AP) - ADDRESS OF FILE NAME
                                0431    711 ;
                                0431    712 ; Output Parameters:
                                0431    713 ;       R0 - Completion status code
                                0431    714 ;
                                0431    715
                           0000 0431    716          .Entry  BOO$WRTSYSPARFILE,^M<>
00000498'EF   02000000 8F   C8  0433    717          BISL    #FAB$M_CIF,FAB$L_FOP+-  ; Set the create if flag in the fab
                                043E    718                  PRM_OUTFAB              ; we don't want multiple versions of this .P
          000004E8'EF   02   C8  043E    719          BISL    #RAB$M_TPT,RAB$L_ROP+-  ; Set the update flag in the rab we are
                                0445    720                  PRM_OUTRAB              ; overwriting an existing file
              FF06 CF   6C   FA  0445    721          CALLG   (AP),BOO$WRTFILE        ; Call routine to write the .PAR file
00000498'EF   02000000 8F   CA  044A    722          BICL    #FAB$M_CIF,FAB$L_FOP+-  ; Clear the create if flag in the fab
                                0455    723                  PRM_OUTFAB              ;
          000004E8'EF   02   CA  0455    724          BICL    #RAB$M_TPT,RAB$L_ROP+-  ; Clear the update flag in the rab
                                045C    725                  PRM_OUTRAB              ;
                           04   045C    726          RET                             ; Return
```

```
                                045D   728          .SBTTL  BOO$SWPCREATE - CREATE SWAP OR PAGING FILE
                                045D   729 ;++
                                045D   730 ; FUNCTIONAL DESCRIPTION:
                                045D   731 ;       BOO$CREATE IS CALLED TO IMPLEMENT THE CREATE COMMAND IN THE SYSGEN
                                045D   732 ;       UTILITY.  THIS COMMAND IS USED TO CREATE DUMP,SWAP AND PAGING FILES.
                                045D   733 ;
                                045D   734 ; INPUT PARAMETERS:
                                045D   735 ;       BOO$GL_SIZE - SIZE OF FILE TO CREATE IN BLOCKS
                                045D   736 ;
                                045D   737 ; OUTPUT PARAMETERS:
                                045D   738 ;       R0 - COMPLETION STATUS CODE
                                045D   739 ;--
                         00FC   045D   740 .ENTRY  BOO$CREATE      ^M<R2,R3,R4,R5,R6,R7>   ; SAVE R2-R7
                                045F   741
    57   00000528'EF    DE     045F   742          MOVAL   SWAPFHCFAB,R7           ; GET FAB ADDRESS
                34 A7    95     0466   743          TSTB    FAB$B_FNS(R7)           ; SEE IF FILE WAS SPECIFIED
                   08    12     0469   744          BNEQ    10$                     ; BRANCH IF IT WAS
    50   00000000'8F    D0     046B   745          MOVL    #<<RMS$_FNM> & ^C<STS$M_SEVERITY>>,R0 ; SET WARNING
                   04    04     0472   746          RET                             ; RETURN
                                0473   747
    55   0000062C'EF    D0     0473   748 10$:     MOVL    BOO$GL_SIZE,R5          ; SET DESIRED SIZE
    56   000005C4'EF    DE     047A   749          MOVAL   SWAPAL[XABNC,R6         ; GET ADDRESS OF NONCONTIG ALLOCATION XAB
00000000'EF    11     E1     0481   750          BBC     #BOOCMD$V_CONTIG, -
                   07            0488   751                  BOO$GL_CMDOPT,20$       ; BR IF THAT'S WHAT'S WANTED
    56   000005A4'EF    DE     0489   752          MOVAL   SWAPAL[XAB,R6           ; GET ADDRESS OF CONTIG ALLOCATION XAB
0000057C'EF    56     D0     0490   753 20$:     MOVL    R6,SWAPFHCXAB+XAB$L_NXT ; SET ADDRESS OF ALLOCATION XAB
00000000'EF    11     E0     0497   754          BBS     #BOOCMD$V_CONTIG, -
                   75            049E   755                  BOO$GL_CMDOPT,CREATE    ; BR IF CONTIGUOUS SPECIFIED
                                049F   756 ;
                                049F   757 ; Try to open file to see if it exists
                                049F   758 ;
04 A7   00020000 8F    CA     049F   759          BICL    #FAB$M_UFO,FAB$L_FOP(R7); CLEAR USER FILE OPEN
                                04A7   760          $OPEN   FAB=(R7)                ; OPEN
      08 A6    20    C8     04B0   761          BISL    #XAB$M_CBT,XAB$B_AOP(R6); SET FOR CONTIGUOUS BEST TRY ALLOCATION
            5D 50    E9     04B4   762          BLBC    R0,CREATE               ; IF ERROR, CREATE FILE
                                04B7   763 ;
                                04B7   764 ; Extend file
                                04B7   765 ;
 51   55    10 A6    C3     04B7   766          SUBL3   XAB$L_ALQ(R6),R5,R1     ; DIFF
                35    13     04BC   767          BEQL    30$                     ; IF THE SA E, THEN NOTHING TO DO
                4B    19     04BE   768          BLSS    50$                     ; IF LESS THEN NEW FILE
      10 A6    51    D0     04C0   769          MOVL    R1,XAB$L_ALQ(R6)
                                04C4   770          $EXTEND FAB=(R7)
            38 50    E9     04CD   771          BLBC    R0,40$
                                04D0   772
 54   000005E4'EF    9E     04D0   773          MOVAB   SWAP_RAB,R4             ; RAB ADDRESS
                                04D7   774          $CONNECT RAB=(R4)
            25 50    E9     04E0   775          BLBC    R0,40$
      38 A4    55    D0     04E3   776          MOVL    R5,RAB$L_BKT(R4)        ; SET LENGTH
                                04E7   777          $WRITE  RAB=(R4)                ; FORCE EOF TO SIZE SPECIFIED
            15 50    E9     04F0   778          BLBC    R0,40$                  ; BRANCH ON ERROR
                                04F3   779 30$:     $CLOSE  FAB=(R7)                ; CLOSE FILE
            09 50    E9     04FC   780          BLBC    R0,40$                  ; EXIT IF ERROR
 50   007CA033 8F    D0     04FF   781          MOVL    #SYSG$_EXTENDED,R0      ; SET STATUS
                71    11     0506   782          BRB     CREATE_SIGNAL           ; OK TO SIGNAL
              0094    31     0508   783 40$:     BRW     CREATE_ERROR            ; EXIT
                                050B   784
```

```
                                050B    785 50$:    $CLOSE  FAB=(R7)
                                0514    786
                 10 A6    55    D0  0514    787 CREATE: MOVL     R5,XAB$L_ALQ(R6)          ; SET ALLOCATION REQUEST SIZE
        04 A7    00020000 8F    C8  0518    788         BISL     #FAB$M_UFO,FAB$L_FOP(R7); SET USER FILE OPEN, SO STV WILL BE CHANNEL
                                0520    789         $CREATE FAB=(R7)                  ; OPEN AND CREATE FILE
                       10 50    E8  0529    790         BLBS     R0,10$                   ; CONTINUE IF NO ERROR
                       0C A7    DD  052C    791         PUSHL    FAB$L_STV(R7)            ; SET FAILURE STATUS VALUE
                 7E    50       07  052F    792         BICL3    #STS$M_SEVERITY,R0,-(SP); SET FAILURE STATUS, CONVERTING TO WARNING
            00000000'GF    02    FB  0533    793         CALLS    #2,G^LIB$SIGNAL          ; SIGNAL THE FAILURE
                           60    11  053A    794         BRB      CREATE_EXIT              ;
                                053C    795
                                053C    796 10$:    $QIOW_S CHAN=FAB$L_STV(R7),-      ; ZERO FIRST LONGWORDS IN FILE
                                053C    797                 FUNC=#IO$_WRITEVBLK,-     ; WRITE VIRUAL BLOCK
                                053C    798                 P1=ZEROES,-               ; ADDRESS OF ZEROES
                                053C    799                 P2=#512,-                 ; ONE PAGE
                                053C    800                 P3=#1                     ; FIRST BLOCK OF FILE
                                0564    801         $DASSGN_S CHAN=FAB$L_STV(R7)      ; DEASSIGN CHANNEL
                       2D 50    E9  056F    802         BLBC     R0,CREATE_ERROR
            50   00001070 8F    D0  0572    803         MOVL     #SHR$_CREATED,R0         ; RETURN SUCCESS
                                0579    804
                                0579    805 CREATE_SIGNAL:
                                0579    806
            000000B0'EF    9F   0579    807         PUSHAB   RSLFNM                   ; SET FILE NAME ADDRESS
        7E  0000035F'EF    9A   057F    808         MOVZBL   RIO_INPNAM+NAM$B_RSL,-(SP) ; SET FILE NAME SIZE
                 51    5E    D0  0586    809         MOVL     SP,R1                    ; GET ADDRESS OF DESCRIPTOR
                       51    DD  0589    810         PUSHL    R1                       ; SET FILE NAME DESCRIPTOR ADDRESS
                       01    DD  058B    811         PUSHL    #1                       ; SET FAO COUNT
            007C0003 8F    C9  058D    812         BISL3    #<SYSG$_FACILITY@16!STS$K_INFO> -
                 7E    50       0593    813                  ,R0,-(SP)                ; MAKE SYSGEN MESSAGE
            00000000'GF    03    FB  0595    814         CALLS    #3,G^LIB$SIGNAL          ; SIGNAL THE ACTION JUST TAKEN
                                059C    815
                                059C    816 CREATE_EXIT:
                 50    0'     D0  059C    817         MOVL     #1,R0                    ; Set sucess
                                059F    818
                                059F    819 CREATE_ERROR:
                 34 A7    94    059F    820         CLRB     FAB$B_FNS(R7)            ; CLEAR FILE NAME LENGTH
                 2C A7    D4    05A2    821         CLRL     FAB$L_FNA(R7)            ; AND FILE NAME ADDRESS
                          04    05A5    822         RET
                                05A6    823
```

```
                        05A6   825          .SBTTL  INSTALL PAGE OR SWAP FILE
                        05A6   826 ;++
                        05A6   827 ;
                        05A6   828 ; Functional Description:
                        05A6   829 ;
                        05A6   830 ;       BOO$INSTALL is called to install a second page or swap file. The
                        05A6   831 ;       specified file is opened, the window pointer is removed from the
                        05A6   832 ;       associated channel control block and moved to the page file control
                        05A6   833 ;       block. The PFL address is loaded into the page file control block
                        05A6   834 ;       vector.
                        05A6   835 ;
                        05A6   836 ; Input Parameters:
                        05A6   837 ;
                        05A6   838 ;       SWAPFHCFAB - Specified file name fields filled in.
                        05A6   839 ;       SWAPFHCXAB - File size is stored in XAB
                        05A6   840 ;       BOO$GL_CMDOPT <BOOCMD$V_PAGEFILE> - Set if page file installation
                        05A6   841 ;
                        05A6   842 ; Output Parameters:
                        05A6   843 ;
                        05A6   844 ;       None
                        05A6   845 ;
                        05A6   846 ; Implicit Output:
                        05A6   847 ;
                        05A6   848 ;       A page file control block describing the designated file is created
                        05A6   849 ;       and loaded into the page file control block vector. (The details
                        05A6   850 ;       can be found in the module header in INITPGFIL.)
                        05A6   851 ;
                        05A6   852 ; Completion Status:
                        05A6   853 ;
                        05A6   854 ;       R0 - Completion status
                        05A6   855 ;
                        05A6   856 ;               All status returns (success and failure codes) are returned
                        05A6   857 ;               from kernel mode by INITPAGSWPFIL (which gets its status
                        05A6   858 ;               from either BOO$INITPAGFIL or BOO$INITSWPFIL).
                        05A6   859 ;
                        05A6   860 ;--
                        05A6   861 BOO$INSTALL::
                  00FC  05A6   862          .WORD   ^M<R2,R3,R4,R5,R6,R7>   ; Entry mask
    57  00000528'EF    DE  05A8   863          MOVAL   SWAPFHCFAB,R7           ; Get base of FAB
04  A7  00020000 8F    C8  05AF   864          BISL    #FAB$M_UFO,FAB$L_FOP(R7); SET USER FILE OPEN, SO STV WILL BE CHANNEL
                        05B7   865
                        05B7   866 10$:     $OPEN   FAB=(R7)                ; Open file for install
              28 50     E8  05C0   867          BLBS    R0,20$                  ; Continue if no error
        00000030'EF     9F  05C3   868          PUSHAB  EXPFNM                  ; Set file name address
    7E  00000367'EF     9A  05C9   869          MOVZBL  R10_INPNAM+NAM$B_ESL,-(SP) ; Set file name size
          51    5E      D0  05D0   870          MOVL    SP,R1                   ; Get address of descriptor
             0C A7      DD  05D3   871          PUSHL   FAB$L_STV(R7)           ; Set status value
                50      DD  05D6   872          PUSHL   R0                      ; Set status
                51      DD  05D8   873          PUSHL   R1                      ; Set file name descriptor address
                01      DD  05DA   874          PUSHL   #1                      ; Set FAO count
        007C1098 8F     DD  05DC   875          PUSHL   #SHR$_OPENIN!<SYSG$_FACILITY@16> ; Set message status
00000000'GF     05      FB  05E2   876          CALLS   #5,G^LIB$SIGNAL         ; Signal the failure
                1B      11  05E9   877          BRB     30$                     ; Return successfully because
                            05EB   878                                          ;  error already signalled
                            05EB   879
                            05EB   880 20$:     $CMKRNL_S INITPAGSWPFIL
             0F 50      E8  05FA   881          BLBS    R0,40$                  ; Branch if success
```

```
                50   DD  05FD    882 25$:    PUSHL   R0                          ; Set failure status
00000000'GF     01   FB  05FF    883         CALLS   #1,G^LIB$SIGNAL             ; Signal the failure
    50  0000'8F   3C  0606    884 30$:    MOVZWL  #SS$_NORMAL,R0              ; Error already signalled
               04   060B    885 35$:    RET
                    060C    886 40$:
                    060C    887 ;**JNL**        BBC     #BOOCMD$V_PAGEFILE,BOO$GL_CMDOPT,35$ ; Nothing else to do if
                    060C    888 ;**JNL**        BBS     #BOOCMD$V_NOCHKPNT,BOO$GL_CMDOPT,60$ ; Nocheckpoint - invali
                    060C    889 ;**JNL**        PUSHL   #1                          ; Full checkpoint processing
                    060C    890 ;**JNL**        PUSHL   BOO$GL_INDEX                ; Page file index
                    060C    891 ;**JNL**        CALLS   #2,CHK$INSPGFIL             ; Protect checkpoints and init the f
                    060C    892 ;**JNL**        BLBS    R0,50$                      ; Success
                    060C    893 ;**JNL**        PUSHL   R0
                    060C    894 ;**JNL**        $CMKRNL_S CLNUPPAGFIL               ; Clean up failed installation
                    060C    895 ;**JNL**        POPL    R0
                    060C    896 ;**JNL** 50$:   RET
                    060C    897 ;**JNL** ;
                    060C    898 ;**JNL** 60$:   PUSHL   BOO$GL_INDEX                ; Page file index
                    060C    899 ;**JNL**        CALLS   #1,CHK$INVPGFL              ; Invalidate any saved checkpoints
                    060C    900 ;**JNL**        BLBC    R0,25$                      ; Go signal the error
                    060C    901 ;**JNL**        RET
                    060C    902 ;
                    060C    903 ; Install a second page or swap file
                    060C    904 ;
                    060C    905
                    060C    906 INITPAGSWPFIL:
              0004  060C    907         .WORD   ^M<R2>                      ; Entry mask
    00000630'EF   DF  060E    908         PUSHAL  BOO$GL_INDEX                ; Returned PFL index
              7E  D4  0614    909         CLRL    -(SP)                       ; Default MAXVBN parameter to default
52  00000528'EF   DE  0616    910         MOVAL   SWAPFHCFAB,R2               ; Get base of FAB
      52  0C A2   CE  061D    911         MNEGL   FAB$L_STV(R2),R2            ; Get channel number and invert
52  00000000'9F   C0  0621    912         ADDL    @#CTL$GL_CCBBASE,R2         ; Compute base of channel
          04 A2   DD  0628    913         PUSHL   CCB$L_WIND(R2)              ; Get window pointer address
7E  00000588'EF   01  C3  062B    914         SUBL3   #1,SWAPFHCXAB+XAB$L_EBK,-(SP) ; Get size of file
  09 00000000'EF   04  E0  0633    915         BBS     #BOOCMD$V_PAGEFILE,BOO$GL_CMDOPT,10$ ; Br if installing pagefile
00000000'GF   04  FB  063B    916         CALLS   #4,G^BOO$INITSWPFIL         ; Call external procedure
              07  11  0642    917         BRB     20$                         ; Join common completion code
                    0644    918 10$:
                    0644    919 ;**JNL**        BBS     #BOOCMD$V_NOCHKPNT,BOO$GL_CMDOPT,15$ ; Is it marked for no c
                    0644    920 ;**JNL**        MOVB    #1,11(SP)                   ; No - don't mark file available yet
00000000'GF   04  FB  0644    921 15$:    CALLS   #4,G^BOO$INITPAGFIL         ; Call external procedure
          23 50   E9  064B    922 20$:    BLBC    R0,30$                      ; Skip next if error
    50  04 A2   D0  064E    923         MOVL    CCB$L_WIND(R2),R0           ; Disconnect file from this process by
00000000'EF   16  0652    924         JSB     MMG$RET_BYT_QUOTA           ;   refund byte quota & clear PID in WCB
    52  04 A2   D0  0658    925         MOVL    CCB$L_WIND(R2),R2           ; Get window control block address
      0E A2   B6  065C    926         INCW    WCB$W_REFCNT(R2)            ; Jimmy ref count so WCB sticks around
52  00000528'EF   DE  065F    927         MOVAL   SWAPFHCFAB,R2               ; Get base of FAB
                    0666    928         $DASSGN_S CHAN = FAB$L_STV(R2)      ; Deassign the channel to unlock the file
               04   0671    929 30$:    RET
                    0672    930 ;
                    0672    931 ;
                    0672    932 ; Clean up after a page file installation has failed
                    0672    933 ;
                    0672    934 CLNUPPAGFIL:
              0038  0672    935         .WORD   ^M<R3,R4,R5>
    53  0C A7   CE  0674    936         MNEGL   FAB$L_STV(R7),R3            ; Get channel number and invert
53  00000000'9F   C0  0678    937         ADDL    @#CTL$GL_CCBBASE,R3         ; Compute base of channel
54  00000630'EF   D0  067F    938         MOVL    BOO$GL_INDEX,R4             ; Page file index
```

```
            55   00000000'FF44    D0  0686    939          MOVL    @MMG$GL_PAGSWPVC[R4],R5 ; Page file control block
                 04 A3   0C A5    D0  068E    940          MOVL    PFL$L_WINDOW(R5),CCB$L_WIND(R3) ; Put window back
                                       0693    941          $DASSGN_S CHAN = FAB$L_STV(R7)  ; Deassign the channel
                                       069E    942          SETIPL  10$                     ; Go to synch and lock pages
   00000000'FF44    00000000'GF    DE  06A5    943          MOVAL   G^MMG$GL_NULLPFL,@MMG$GL_PAGSWPVC[R4] ; Free the vector slot
                        50    55    D0  06B1    944          MOVL    R5,R0                   ; Address of PFL block
                     51    08 A5    3C  06B4    945          MOVZWL  PFL$W_SIZE(R5),R1       ; PFL size
                     51    14 A5    C0  06B8    946          ADDL2   PFL$L_BITMAPSIZ(R5),R1  ; Add the bitmap
                     00000000'GF    16  06BC    947          JSB     G^EXE$DEANONPGDSIZ      ; Deallocate it all
                                       06C2    948          SETIPL  #0
                                   04  06C5    949          RET
                                       06C6    950 ;
                                  0008 06C6    951 10$:     .WORD   IPL$_SYNCH
```

```
                                    06C8            .SBTTL  INSTALL/CREATE ACTION ROUTINES
                                    06C8
                                    06C8    BOO$SETPGFL::                                   ; SET PAGEFILE OPTION
                              0000  06C8            .WORD   0                               ;
        00 00000000'EF    04    E2  06CA            BBSS    #BOOCMD$V_PAGEFILE,BOO$GL_CMDOPT,10$
                                04  06D2    10$:    RET                                     ;
                                    06D3
                                    06D3    BOO$CRENCONTIG::                                ; CREATE NON-CONTIGUOUS FILE
                              0000  06D3            .WORD   0
           00000000'EF    20    C8  06D5            BISL    #BOOCMD$M_NONCONTIG,BOO$GL_CMDOPT  ; SET OPTION BIT
     00000000'EF    00020000 8F CA  06DC            BICL    #BOOCMD$M_CONTIG,BOO$GL_CMDOPT  ; CLEAR OPTION BIT
                                04  06E7            RET                                     ;
                                    06E8
                                    06E8    BOO$CRECONTIG::                                 ; CREATE CONTIGUOUS FILE
                              0000  06E8            .WORD   0
           00000000'EF    20    CA  06EA            BICL    #BOOCMD$M_NONCONTIG,BOO$GL_CMDOPT ; CLEAR OPTION BIT
     00000000'EF    00020000 8F C8  06F1            BISL    #BOOCMD$M_CONTIG,BOO$GL_CMDOPT  ; SET OPTION BIT
                                04  06FC            RET                                     ;
                                    06FD
                                    06FD    BOO$FILESIZE::                                  ; SET FILE SIZE
                              0000  06FD            .WORD   0                               ;
        0000062C'EF    1C AC    D0  06FF            MOVL    TPA$L_NUMBER(AP),BOO$GL_SIZE    ;
                                04  0707            RET                                     ;
                                    0708
                                    0708    BOO$SETFILNAM::                                 ;
                              0000  0708            .WORD   0                               ;
        51   00000528'EF        9E  070A            MOVAB   SWAPFHCFAB,R1                   ;
          34 A1    10 AC        90  0711            MOVB    TPA$L_TOKENCNT(AP),FAB$B_FNS(R1) ; SET FILE NAME LENGTH
          2C A1    14 AC        D0  0716            MOVL    TPA$L_TOKENPTR(AP),FAB$L_FNA(R1) ; AND FILE NAME ADDRESS
                                04  071B            RET                                     ;
                                    071C
                                    071C    BOO$NOCHKPNT::
                              0000  071C            .WORD   0                               ; PAGE FILE DOES NOT SUPPORT CHECKPOINT/REST
        00 00000000'EF    12    E2  071E            BBSS    #BOOCMD$V_NOCHKPNT,BOO$GL_CMDOPT,10$
                                04  0726    10$:    RET
```

```
                     0727    989 .SBTTL RMS DEFINITIONS FOR /OUTPUT= QUALIFIER
                     0727    990
                  0000063B   991 .Psect  PAGED_DATA      rd,wrt,noexe,quad
                     063B    992 .ALIGN  LONG
                     063C    993
                     063C    994 ;       Output File Data
                     063C    995
                0000 063C    996 RIO$GW_OUTLEN::         .WORD   0                ;LENGTH OF STRING
                0000 063E    997 RIO$AB_OUTBUF::         .LONG   BUFFER_SIZE      ;OUTPUT BUFFER
00000646'       0642    998                             .LONG   RIO$AB_BUFFER
00000746        0646    999 RIO$AB_BUFFER::         .BLKB   BUFFER_SIZE
                0746   1000 ;
                0746   1001 ;       Output Device Data
                0746   1002 ;
                0746   1003 .ALIGN LONG
                0748   1004 RIO_OUTFAB2::   $FAB    RAT=CR
                0798   1005 RIO_OUTRAB2::   $RAB    FAB=RIO_OUTFAB2
                07DC   1006 RIO_OUTFAB3:    $FAB    RAT=CR
                082C   1007 RIO_OUTRAB3:    $RAB    FAB=RIO_OUTFAB3
                0870   1008
                0870   1009 ;        Input File Data
                0870   1010
00000100        0870   1011 RIO$GL_INBUF_SZ::        .LONG   BUFFER_SIZE
00000974        0874   1012 RIO$AB_INBUFFER::        .BLKB   BUFFER_SIZE
                0974   1013
                0974   1014 ; Input Device Data
                0974   1015
                0974   1016         .ALIGN  LONG
                0974   1017 RIO_INFAB2::   $FAB    FNM=<INFILE:>
                09C4   1018 RIO_INRAB2::   $RAB    FAB=RIO_INFAB2,-
                09C4   1019                                 UBF=RIO$AB_INBUFFER,-
                09C4   1020                                 USZ=RIO$GL_INBUF_SZ,PBF=PROMPT_IN, -
                09C4   1021                                 ROP=<PMT,CVT>,PSZ=8
20 3E 45 43 49 56 45 44 0A08   1022 PROMPT_IN:      .ASCII  /DEVICE> /
                0A10   1023
                   09   0A10   1024 INNAM_SIZE:      .BYTE   INPNAMSZ
          000001C4'     0A11   1025 INNAM_ADDR:      .LONG   INPNAM
                0A15   1026
          4D 4F 43 2E   0A15   1027 COM:     .ASCII  /.COM/
          53 49 4C 2E   0A19   1028 LIS:     .ASCII  /.LIS/
```

```
                        00000727  1030 .Psect  PAGED_CODE      rd,nowrt,exe,long
                            0727  1031
                            0727  1032 ;
                            0727  1033 ; The following are called as TPARSE action routines from SYSBOOCMD
                            0727  1034 ;
                            0727  1035
                     0000   0727  1036 .ENTRY  BOO$INPUT_FILE, ^M<>
                            0729  1037
            14 AC    D0     0729  1038         MOVL    TPA$L_TOKENPTR(AP),-
00000A11'EF                 072C  1039                 INNAM_ADDR              ; Save pointer to input file
            10 AC    F6     0731  1040         CVTLB   TPA$L_TOKENCNT(AP),-
00000A10'EF                 0734  1041                 INNAM_SIZE              ; Save length of file name
00004000 8F  C8     0739  1042         BISL    #BOOCMD$M_INPUT,-
00000000'EF                 073F  1043                 BOO$GL_CMDOPT           ; Set flag for /INPUT
                     04     0744  1044         RET                             ; Return
                            0745  1045
                     0000   0745  1046 .ENTRY  BOO$OUTPUT_FILE, ^M<>
                            0747  1047
            14 AC    D0     0747  1048         MOVL    TPA$L_TOKENPTR(AP),-
00000000'EF                 074A  1049                 BOO$GL_FILEADDR         ; Save pointer to output file
            10 AC    F6     074F  1050         CVTLB   TPA$L_TOKENCNT(AP),-
00000000'EF                 0752  1051                 BOO$GB_FILELEN          ; Save length of file name
00002000 8F  C8     0757  1052         BISL    #BOOCMD$M_OUTPUT,-
00000000'EF                 075D  1053                 BOO$GL_CMDOPT           ; Set flag for /INPUT
                     04     0762  1054         RET                             ; Return
                            0763  1055
                     0000   0763  1056 .ENTRY  BOO$RESET_IO, ^M<>
                            0765  1057
00000778'EF  00000A19'EF    DE   0765  1058         MOVAL   LIS,RIO_OUTFAB2+FAB$L_DNA       ; Set .lis default
      0000077D'EF    04     90   0770  1059         MOVB    #4,RIO_OUTFAB2+FAB$B_DNS        ; Set size of .lis
                            0777  1060
      00000000'EF    0A     90   0777  1061         MOVB    #OUTPNAMSZ,BOO$GB_FILELEN       ; Set default length
00000000'EF  000001CD'EF    DE   077E  1062         MOVAL   OUTPNAM,BOO$GL_FILEADDR         ; Default output is SYS$OUTPUT
      00000A10'EF    09     90   0789  1063         MOVB    #INPNAMSZ,INNAM_SIZE            ; Set default length
00000A11'EF  000001C4'EF    DE   0790  1064         MOVAL   INPNAM,INNAM_ADDR               ; Default input is SYS$INPUT
                     04     079B  1065         RET                             ; Return
                            079C  1066
                     0000   079C  1067 .ENTRY  BOO$RESET_COMMAND, ^M<>
                            079E  1068
                            079E  1069 ; Command SHOW/CONFIG/COMMAND_FILE was originally SAVE <file-spec>
                            079E  1070
00000778'EF  00000A15'EF    DE   079E  1071         MOVAL   COM,RIO_OUTFAB2+FAB$L_DNA       ; Set up RMS default extension
      0000077D'EF    04     90   07A9  1072         MOVB    #4,RIO_OUTFAB2+FAB$B_DNS        ; Set up extension size
   00 00000000'EF    02     E2   07B0  1073         BBSS    #BOOCMD$V_SAVE,BOO$GL_CMDOPT,10$ ; Set SAVE command flag
                     04     07B8  1074 10$:     RET
                            07B9  1075
```

```
                                07B9  1077 ;
                                07B9  1078 ; These routines are called via a BSBW from TPARSE action routines
                                07B9  1079 ;
                                07B9  1080
                                07B9  1081 BOO$OPEN_OUTPUT_2::                          ; Open option output file
                                07B9  1082
        4D 00000000'EF    0D    E1  07B9  1083          BBC     #BOOCMD$V_OUTPUT,BOO$GL_CMDOPT,10$ ; Branch if no /OUTPUT
        12 00000000'EF    02    E1  07C1  1084          BBC     #BOOCMD$V_SAVE,BOO$GL_CMDOPT,5$    ; Branch if not ".com"
00000778'EF    00000A15'EF    DE  07C9  1085          MOVAL   COM,RIO_OUTFAB2+FAB$L_DNA          ; Set .com default
      0000077D'EF    04    90  07D4  1086          MOVB    #4,RIO_OUTFAB2+FAB$B_DNS           ; Set size of .com
                                07DB  1087
            00000000'EF    90  07DB  1088 5$:      MOVB    BOO$GB_FILELEN,-
            0000077C'EF        07E1  1089                  RIO_OUTFAB2+FAB$B_FNS   ; Set up RMS data of size
            00000000'EF    D0  07E6  1090          MOVL    BOO$GL_FILEADDR,-
            00000774'EF        07EC  1091                  RIO_OUTFAB2+FAB$L_FNA   ;  and addr. of file spec.
                               07F1  1092          $CREATE FAB = RIO_OUTFAB2       ; Create file
                  0D 50    E9  07FE  1093          BLBC    R0,10$                  ; Branch on error
                               0801  1094
                               0801  1095 7$:
                               0801  1096          $CONNECT RAB = RIO_OUTRAB2      ;  and open file
                            05 080E  1097 10$:     RSB
                               080F  1098
                               080F  1099 BOO$OPEN_INPUT_2::
                            0E 080F  1100          BBC     #BOOCMD$V_INPUT,-
          33 00000000'EF        0811  1101                  BOO$GL_CMDOPT,10$       ; Branch if no /INPUT
             00000A10'EF    90  0817  1102          MOVB    INNAM_SIZE,-
             000009A8'EF        081D  1103                  RIO_INFAB2+FAB$B_FNS    ; Set up RMS data of size
             00000A11'EF    D0  0822  1104          MOVL    INNAM_ADDR,-
             000009A0'EF        0828  1105                  RIO_INFAB2+FAB$L_FNA    ; and addr. of file spec.
                               082D  1106          $OPEN   FAB=RIO_INFAB2          ; and open file
                  0D 50    E9  083A  1107          BLBC    R0,10$                  ; branch if error
                               083D  1108
                               083D  1109          $CONNECT RAB=RIO_INRAB2         ; Establish record stream
                            05 084A  1110 10$:     RSB
                               084B  1111
```

```
                        084B  1113 .SBTTL  BOO$SET_OUTPUT - Open file for SET/OUTPUT
                        084B  1114
                   0000 084B  1115 .Entry  BOO$SET_OUTPUT,^M<>             ; Open option output file
                        084D  1116
                 0F  E1 084D  1117         BBC     #BOOCMD$V_SETOUTPUT,-
10 00000000'EF          084F  1118                 BOO$GL_CMDOPT,10$       ; Has this file been open?
                        0855  1119         $CLOSE  FAB = RIO_OUTFAB3       ; Close it first then
               75 50    E9 0862  1120         BLBC    R0,30$                  ; Branch if failure
                        0865  1121
   00000A19'EF          DE 0865  1122 10$:    MOVAL   LIS,-
   0000080C'EF          086B  1123                 RIO_OUTFAB3+FAB$L_DNA   ; Set .lis default
                 04     90 0870  1124         MOVB    #4,-
   00000811'EF          0872  1125                 RIO_OUTFAB3+FAB$B_DNS   ; Set size of .lis
                        0877  1126
   00000000'EF          90 0877  1127         MOVB    BOO$GB_FILELEN,-
   00000810'EF          087D  1128                 RIO_OUTFAB3+FAB$B_FNS   ; Set up RMS data of size
   00000000'EF          D0 0882  1129         MOVL    BOO$GL_FILEADDR,-
   00000808'EF          0888  1130                 RIO_OUTFAB3+FAB$L_FNA   ;  and addr. of file spec.
                        088D  1131
                        088D  1132         $CREATE FAB = RIO_OUTFAB3       ; Create file
               3D 50    E9 089A  1133         BLBC    R0,30$                  ; Branch on error
                        089D  1134
   00010000 8F          CA 089D  1135         BICL    #BOOCMD$M_TERMINAL,-
   00000000'EF          08A3  1136                 BOO$GL_CMDOPT           ; ASSUME NOT A TERMINAL DEVICE
   00000000'8F  E1      08A8  1137         BBC     #DEV$V_TRM,-
   0000081C'EF          08AE  1138                 RIO_OUTFAB3+FAB$L_DEV,-
                 0B     08B3  1139                 20$                     ; BRANCH IF NOT
   00010000 8F          C8 08B4  1140         BISL    #BOOCMD$M_TERMINAL,-
   00000000'EF          08BA  1141                 BOO$GL_CMDOPT           ; SET AS A TERMINAL DEVICE
                        08BF  1142
                        08BF  1143 20$:     $CONNECT RAB = RIO_OUTRAB3      ;  and open file
               0B 50    E9 08CC  1144         BLBC    R0,30$                  ; Branch on error
                        08CF  1145
   00008000 8F          C8 08CF  1146         BISL    #BOOCMD$M_SETOUTPUT,-
   00000000'EF          08D5  1147                 BOO$GL_CMDOPT           ; Set output open
                 04     08DA  1148 30$:     RET
                        08DB  1149
```

```
                                08DB  1151
                                08DB  1152 RIO$Output_line::
                                08DB  1153
                                08DB  1154 ; Output Routine: Use special FAB if SET/OUTPUT or /OUTPUT was specified,
                                08DB  1155 ; RIO$OUTPUT if not.
                                08DB  1156
      25 00000000'EF    0D    E1  08DB  1157          BBC     #BOOCMD$V_OUTPUT,BOO$GL_CMDOPT,10$; /OUTPUT = SPECIFIED?
000007BA'EF    0000063C'EF    B0  08E3  1158          MOVW    RIO$GW_OUTLEN,RIO_OUTRAB2+RAB$W_RSZ
000007C0'EF    00000646'EF    DE  08EE  1159          MOVAL   RIO$AB_BUFFER,RIO_OUTRAB2+RAB$L_RBF
                                08F9  1160          $PUT    RAB=RIO_OUTRAB2
                         3F     11  0906  1161          BRB     30$
                                0908  1162
      25 00000000'EF    0F    E1  0908  1163 10$:     BBC     #BOOCMD$V_SETOUTPUT,BOO$GL_CMDOPT,20$; SET /OUTPUT= SPECIFIED?
0000084E'EF    0000063C'EF    B0  0910  1164          MOVW    RIO$GW_OUTLEN,RIO_OUTRAB3+RAB$W_RSZ
00000854'EF    00000646'EF    DE  091B  1165          MOVAL   RIO$AB_BUFFER,RIO_OUTRAB3+RAB$L_RBF
                                0926  1166          $PUT    RAB=RIO_OUTRAB3
                         12     11  0933  1167          BRB     30$
                                0935  1168
                                0935  1169 20$:
          7E    0000063C'EF    3C  0935  1170          MOVZWL  RIO$GW_OUTLEN,-(SP)     ; Length
                00000646'EF    DF  093C  1171          PUSHAL  RIO$AB_BUFFER           ; Address
              F71C CF    02    FB  0942  1172          CALLS   #2,RIO$OUTPUT           ; Output to SYS$OUTPUT
                   09 50    E8  0947  1173 30$:     BLBS    R0,40$                  ; Branch if OK
                      50    DD  094A  1174          PUSHL   R0
          00000000'GF    01    FB  094C  1175          CALLS   #1,G^LIB$SIGNAL         ; Signal
                                0953  1176
                         05     0953  1177 40$:     RSB                             ; Return
                                0954  1178
                                0954  1179
                                0954  1180          .END
```

```
$$.TAB                  = 000009C4 R      02
$$.TABEND               = 00000A08 R      02
$$.TMP                  = 44000000
$$.TMP1                 = 00000001
$$.TMP2                 = 000000CF
$$.TMPX                 = 00000000 R      04
$$.TMPX1                = 00000007
$$T1                    = 00000001
BOO$AB_PRMBUF             ********    X   02
BOO$A_PRMBLK              ********    X   03
BOO$CREATE                0000045D RG       03
BOO$CRECONTIG             000006E8 RG       03
BOO$CRENCONTIG            000006D3 RG       03
BOO$EXEOPEN               000001E1 RG       03
BOO$FILCLOSE              000002B6 RG       03
BOO$FILESIZE              000006FD RG       03
BOO$FILOPEN               000001F3 RG       03
BOO$FILOPENW              000001D4 RG       03
BOO$GB_FILELEN            ********    X   03
BOO$GL_CMDOPT             ********    X   03
BOO$GL_FILEADDR           ********    X   03
BOO$GL_INDEX              00000630 R        02
BOO$GL_SIZE               0000062C R        02
BOO$GQ_CMDESC             ********    X   03
BOO$INITPAGFIL            ********    X   03
BOO$INITSWPFIL            ********    X   03
BOO$INPUT_FILE            00000727 RG       03
BOO$INSTALL               000005A6 RG       03
BOO$NOCHKPNT              0000071C RG       03
BOO$OPEN_INPUT_2          0000080F RG       03
BOO$OPEN_OUTPUT_2         000007B9 RG       03
BOO$OUTPUT_FILE           00000745 RG       03
BOO$READFILE              000002C7 RG       03
BOO$READPROMPT            000000CF RG       03
BOO$RESET_COMMAND         0000079C RG       03
BOO$RESET_IO              00000763 RG       03
BOO$SETFILNAM             00000708 RG       03
BOO$SETPGFL               000006C8 RG       03
BOO$SET_OUTPUT            0000084B RG       03
BOO$UFOOPEN               00000259 RG       03
BOO$WRITEFILE             00000307 RG       03
BOO$WRTFILE               00000350 RG       03
BOO$WRTSYSPARFILE         00000431 RG       03
BOOCMD$M_CONTIG         = 00020000
BOOCMD$M_INPUT          = 00004000
BOOCMD$M_NONCONTIG      = 00000020
BOOCMD$M_OUTPUT         = 00002000
BOOCMD$M_SETOUTPUT      = 00008000
BOOCMD$M_TERMINAL       = 00010000
BOOCMD$V_CONTIG         = 00000011
BOOCMD$V_INPUT          = 0000000E
BOOCMD$V_NOCHKPNT       = 00000012
BOOCMD$V_OUTPUT         = 0000000D
BOOCMD$V_PAGEFILE       = 00000004
BOOCMD$V_SAVE           = 00000002
BOOCMD$V_SETOUTPUT      = 0000000F
BUF                     = 0000000C
BUFFER_SIZE             = 00000100
CCB$L_WIND              = 00000004
CLNUPPAGFIL               00000672 R        03
COM                       00000A15 R        02
CREATE                    00000514 R        03
CREATE_ERROR              0000059F R        03
CREATE_EXIT               0000059C R        03
CREATE_SIGNAL             00000579 R        03
CTL$GL_CCBBASE            ********    X   03
DEFNAM                    00000000 R        02
DEV$V_TRM                 ********    X   03
EXE$DEANONPGDSIZ          ********    X   03
EXE$GT_STARTUP            ********    X   03
EXEDEF                    000001E2 R        02
EXEDEFSIZ               = 0000000F
EXPFNM                    00000030 R        02
FAB$B_DNS               = 00000035
FAB$B_FAC               = 00000016
FAB$B_FNS               = 00000034
FAB$C_BID               = 00000003
FAB$C_BLN               = 00000050
FAB$C_FIX               = 00000001
FAB$C_SEQ               = 00000000
FAB$C_VAR               = 00000002
FAB$L_ALQ               = 00000010
FAB$L_DEV               = 00000040
FAB$L_DNA               = 00000030
FAB$L_FNA               = 0000002C
FAB$L_FOP               = 00000004
FAB$L_STV               = 0000000C
FAB$M_CIF               = 02000000
FAB$M_PUT               = 00000001
FAB$M_UFO               = 00020000
FAB$V_BIO               = 00000005
FAB$V_CHAN_MODE         = 00000002
FAB$V_CR                = 00000001
FAB$V_CTG               = 00000014
FAB$V_FILE_MODE         = 00000004
FAB$V_GET               = 00000001
FAB$V_LNM_MODE          = 00000000
FAB$V_PUT               = 00000000
FAB$V_SUP               = 00000002
FAB$V_TRN               = 00000004
FAB$W_GBC               = 00000048
FILNAMSIZ               = 00000080
FILOPEN                   000001FE R        03
INITPAGSWPFIL             0000060C R        03
INNAM_ADDR                00000A11 R        02
INNAM_SIZE                00000A10 R        02
INPNAM                    000001C4 R        02
INPNAMSZ                = 00000009
INP_OPEN                  000001F1 R        02
IO$_WRITEVBLK             ********    X   03
IPL$_SYNCH              = 00000008
LIB$SIGNAL                ********    X   03
LIS                       00000A19 R        02
MMG$GL_NULLPFL            ********    X   03
```

```
MMG$GL_PAGSWPVC          ********    X  03
MMG$RET_BYT_QUOTA        ********    X  03
NAM$B_ESL               = 0000000B
NAM$B_ESS               = 0000000A
NAM$B_NOP               = 00000008
NAM$B_RSL               = 00000003
NAM$B_RSS               = 00000002
NAM$C_BID               = 00000002
NAM$C_BLN               = 00000060
NAM$L_ESA               = 0000000C
NAM$L_RSA               = 00000004
NOREAD                   0000016B R     03
OUTFNM                   00000634 R     02
OUTNAM                   00000130 R     02
OUTPNAM                  000001CD R     02
OUTPNAMSZ               = 0000000A
OUTP_OPEN                000001F2 R     02
OUTSIZ                  = 00000007
PFL$L_BITMAPSIZ         = 00000014
PFL$L_WINDOW            = 0000000C
PFL$W_SIZE              = 00000008
PR$_IPL                  ********    X  03
PRM$B_POS               = 00000015
PRM$B_SIZE              = 00000014
PRM$C_LENGTH            = 00000032
PRM$L_ADDR              = 00000000
PRM$L_FLAGS             = 00000010
PRM$T_NAME              = 00000016
PRM$V_ASCII             = 00000010
PRM$V_NEG               = 0000000C
PRMDEF                   000001D7 R     02
PRMDEFSIZ               = 0000000B
PRM_INFAB                00000400 R     02
PRM_INRAB                00000450 R     02
PRM_OUTFAB               00000494 R     02
PRM_OUTRAB               000004E4 R     02
PROMPT                  = 00000004
PROMPT_IN                00000A08 R     02
PUTERROR                 ********    X  03
RAB$B_PSZ               = 00000034
RAB$B_RAC               = 0000001E
RAB$C_BID               = 00000001
RAB$C_BLN               = 00000044
RAB$C_KEY               = 00000001
RAB$C_SEQ               = 00000000
RAB$L_BKT               = 00000038
RAB$L_CTX               = 00000018
RAB$L_PBF               = 00000030
RAB$L_RBF               = 00000028
RAB$L_ROP               = 00000004
RAB$L_UBF               = 00000024
RAB$M_TPT               = 00000002
RAB$V_CVT               = 0000001A
RAB$V_PMT               = 0000001E
RAB$W_RSZ               = 00000022
RAB$W_USZ               = 00000020
RHBUF                    000001B0 R     02
RIO$AB_BUFFER            00000646 RG    02
RIO$AB_INBUFFER          00000874 RG    02
RIO$AB_OUTBUF            0000063E RG    02
RIO$GL_INBUF_SZ          00000870 RG    02
RIO$GW_OUTLEN            0000063C RG    02
RIO$INPUT                00000000 RG    03
RIO$OUTPUT               00000063 RG    03
RIO$OUTPUTC              00000053 RG    03
RIO$OUTPUT_LINE          000008DB RG    03
RIO_INFAB2               00000974 RG    02
RIO_INPFAB               0000030C RG    02
RIO_INPNAM               0000035C RG    02
RIO_INPRAB               000003BC RG    02
RIO_INRAB2               000009C4 RG    02
RIO_OUTFAB               00000218 RG    02
RIO_OUTFAB2              00000748 RG    02
RIO_OUTFAB3              000007DC R     02
RIO_OUTNAM               00000268 RG    02
RIO_OUTRAB               000002C8 RG    02
RIO_OUTRAB2              00000798 RG    02
RIO_OUTRAB3              0000082C R     02
RMS$_EOF                 ********      X  03
RMS$_FNM                 ********      X  03
RSLFNM                   000000B0 R     02
SHR$_CREATED            = 00001070
SHR$_OPENIN             = 00001098
SIZE                    = 00000008
SS$_NORMAL               ********      X  03
STS$K_INFO              = 00000003
STS$M_SEVERITY          = 00000007
SWAPALLXAB               000005A4 R     02
SWAPALLXABNC             000005C4 R     02
SWAPFHCFAB               00000528 R     02
SWAPFHCXAB               00000578 R     02
SWAP_RAB                 000005E4 R     02
SWPDEFNAM                00000628 R     02
SWPDEFNAMSZ             = 00000004
SYS$CLOSE                ********    GX  03
SYS$CMKRNL               ********    GX  03
SYS$CONNECT              ********    GX  03
SYS$CREATE               ********    GX  03
SYS$DASSGN               ********    GX  03
SYS$EXTEND               ********    GX  03
SYS$GET                  ********    GX  03
SYS$OPEN                 ********    GX  03
SYS$PUT                  ********    GX  03
SYS$QIOW                 ********    GX  03
SYS$READ                 ********    GX  03
SYS$WRITE                ********    GX  03
SYSG$_CREPARFIL         = 007C80FA
SYSG$_EXTENDED          = 007CA033
SYSG$_FACILITY          = 0000007C
SYSG$_INVPARFIL         = 007C8102
TPA$L_NUMBER            = 0000001C
TPA$L_TOKENCNT          = 00000010
TPA$L_TOKENPTR          = 00000014
WCB$W_REFCNT            = 0000000E
```

```
XAB$B_AID                       = 00000017
XAB$B_ALN                       = 00000009
XAB$B_AOP                       = 00000008
XAB$B_BKZ                       = 00000016
XAB$C_ALL                       = 00000014
XAB$C_ALLLEN                    = 00000020
XAB$C_FHC                       = 0000001D
XAB$C_FHCLEN                    = 0000002C
XAB$C_LBN                       = 00000002
XAB$L_ALQ                       = 00000010
XAB$L_EBK                       = 00000010
XAB$L_LOC                       = 0000000C
XAB$L_NXT                       = 00000004
XAB$M_CBT                       = 00000020
XAB$V_CBT                       = 00000005
XAB$V_CTG                       = 00000007
XAB$V_HRD                       = 00000000
XAB$W_DEQ                       = 00000014
XAB$W_RFI0                      = 00000018
XAB$W_RFI2                      = 0000001A
XAB$W_RFI4                      = 0000001C
XAB$W_VOL                       = 0000000A
ZEROES                            000001F3 R      02
```

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
PAGED_DATA	00000A1D (2589.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
PAGED_CODE	00000954 (2388.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
$RMSNAM	00000007 (7.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:00.42
Command processing	131	00:00:00.64	00:00:02.48
Pass 1	513	00:00:23.24	00:00:56.00
Symbol table sort	0	00:00:02.15	00:00:05.14
Pass 2	251	00:00:05.05	00:00:09.87
Symbol table output	33	00:00:00.23	00:00:00.38
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	968	00:00:31.41	00:01:14.31

```
The working set limit was 1800 pages.
121882 bytes (239 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1506 non-local and 58 local symbols.
1180 source lines were read in Pass 1, producing 49 object records in Pass 2.
58 pages of virtual memory were used to define 51 macros.
```

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

Macro library name                                  Macros defined
------------------                                  --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1                           1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                               9
_$255$DUA28:[SYSLIB]STARLET.MLB;2                           38
TOTALS (all libraries)                                      48

1863 GETS were required to define 48 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:RMSCONIO/OBJ=OBJ$:RMSCONIO MSRC$:RMSCONIO/UPDATE=(ENH$:RMSCONIO)+EXECML$/LIB+LIB$:BOOTS.MLB/LIB
```

READLBN
LIS

RTFILREAD
LIS

SHARE
LIS

QUSS
LIS

RWBTDRIVR
LIS

MBBTDRIVR
LIS

SCSLOADER
LIS

RMSCONIO
LIS

READDRIV
LIS

PABTDRIVR
LIS

PUBTDRIVR
LIS

PUTERROR
LIS

READPRMPT
LIS