



```
RRRRRRR      MM      MM      SSSSSSSS      CCCCCCCC      000000      NN      NN      IIIIII      000000
RRRRRRR      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      SS      CC      00      00      NNNN      NN      II      00      00
RR      RR      MM      MM      SS      CC      00      00      NNNN      NN      II      00      00
RRRRRRR      MM      MM      SSSSSS      CC      00      00      NN      NN      II      00      00
RRRRRRR      MM      MM      SSSSSS      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      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      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      IIIIII      SSSSSSSS
LLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLL      IIIIII      SSSSSSSS
```

(1)	107	DECLARATIONS
(1)	282	RIOSINPUT - RMS INPUT INTERFACE
(1)	331	RIOSOUTPUT - RMS OUTPUT INTERFACE ROUTINE
(1)	397	BOOSREADPROMPT - Prompt and read input string
(1)	491	BOOSFIOPEN - Routine to open input file
(1)	533	BOOSFUOPEN - Routine to open input file for user access
(1)	563	BOOSFILCLOSE - Routine to close currently open file
(1)	578	BOOSREADFILE - Read parameter file
(1)	605	BOOSWRITEFILE - Write blocks to file
(2)	636	BOOSWRTFILE - Routine to write parameter file
(3)	706	BOOSWRTSYSPARFILE - Routine to write the system parameter file
(3)	728	BOOSWPCREATE - CREATE SWAP OR PAGING FILE
(3)	825	INSTALL PAGE OR SWAP FILE
(3)	953	INSTALL/CREATE ACTION ROUTINES
(3)	989	RMS DEFINITIONS FOR /OUTPUT= QUALIFIER
(3)	1113	BOOSSET_OUTPUT - Open file for SET/OUTPUT

```
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$V_TERMINAL for /OUTPUT=.
0000 43 :
0000 44 : V03-017 WHM0006 Bill Matthews 26-Jun-1984
0000 45 : Added RIO_OUTFAB3 and RIO_OUTRAB3 for use by SET/OUTPUT so it
0000 46 : won't conflict with /OUTPUT=.
0000 47 :
0000 48 : V03-016 WHM0005 Bill Matthews 25-Apr-1984
0000 49 : Added an initial allocation size for parameter file of 15 blocks.
0000 50 :
0000 51 : V03-015 WHM0004 Bill Matthews 04-Apr-1984
0000 52 : Added routine BOO$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 SWAPFHC FAB 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          $$HRDEF          : SHARED MESSAGE DEFINITIONS
0000 122          $$STSDEF         : STATUS MESSAGE DEFINITIONS
0000 123          $$SYSGMSGDEF     : SYSGEN MESSAGE DEFINITIONS
0000 124          $TPADEF          : DEFINE TPARSE SYMBOLS
0000 125          $WCBDEF          : WINDOW CONTROL BLOCK
0000 126
0000 127          :
0000 128          : EQUATED SYMBOLS:
0000 129          :
00000100 0000 130          BUFFER_SIZE=256          : Buffer size for /OUTPUT= & /INPUT=
00000080 0000 131          FILNAMSIZ=128
0000 132
0000 133          :
0000 134          : OWN STORAGE:
0000 135          :
0000 136
00000000 0000 137          .Psect PAGED_DATA          rd,wrt,noexe,quad
0000 138
53 49 4C 2E 0000 139 DEFNAM: .ASCII /.LIS/          : INITIAL DEFAULT IS LISTING
00000030 0004 140          .BLKB 48-4          : DEFAULT NAME GOES HERE
000000B0 0030 141 EXPFNM: .BLKB FILNAMSIZ          : EXPANDED NAME HERE
00000130 00B0 142 RSLFNM: .BLKB FILNAMSIZ          : RESULTANT NAME HERE
000001B0 0130 143 OUTNAM: .BLKB FILNAMSIZ          : OUTPUT FILE NAME HERE
000001C4 01B0 144 RHBUF: .BLKB 20          : RECORD HEADER BUFFER
01C4 145
54 55 50 4E 49 24 53 59 53 01C4 146 INPNAM: .ASCII /SYSS$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 /SYSS$OUTPUT/          : OUTPUT NAME STRING
0000000A 01D7 150 OUTPNAMSZ=-OUTPNAM          : SIZE OF OUTPUT STRING
01D7 151
52 41 50 2E 54 4C 55 41 46 45 44 01D7 152 PRMDEF: .ASCII /DEFAULT.PAR/          : DEFAULT NAME STRING
0000000B 01E2 153 PRMDEFSIZ=-PRMDEF          : STRING LENGTH
01E2 154
2E 3A 4D 45 54 53 59 53 24 53 59 53 01E2 155 EXEDEF: .ASCII /SYSS$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
0268 180 RSA=OUTNAM,- ;
0268 181 RSS=FILNAMSIZ,- ;
0268 182 ESA=OUTNAM,- ; EXPANDED AND RESULT IN SAME PLACE
0268 183 ESS=FILNAMSIZ,- ;
02C8 184 RIO_OUTRAB:: ;
02C8 185 $RAB
02C8 186 RAC=SEQ,- ; RECORD ACCESS IS SEQUENTIAL
02C8 187 RHB=RHBUF,- ; RECORD HEADER BUFFER
02C8 187 FAB=RIO_OUTFAB ; ASSOCIATED FAB
030C 188
030C 189 RIO_INPFAB:: ;
030C 190 $FAB
030C 191 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
035C 198 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
03BC 204 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 $FAB
0400 213 FAC=<GET,BIO>,- ; READ ACCESS
0400 214 DNA=PRMDEF,- ; DEFAULT NAME
0400 215 DNS=PRMDEF$SIZ,- ; DEFAULT NAME SIZE
0400 216 NAM=RIO_INPNAM ; NAME BLOCK
0450 217 PRM_INRAB: ;
0450 218 $RAB
0450 218 FAB=PRM_INFAB ; ASSOCIATED FAB
0494 219 :
0494 220 : PARAMETER FILE OUTPUT RAB/RAB
0494 221 :
0494 222 PRM_OUTFAB: ;
0494 223 $FAB
0494 223 FAC=<PUT,TRN>,- ;
0494 224 DNA=PRMDEF,- ; DEFAULT NAME
0494 225 DNS=PRMDEF$SIZ,- ; 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=SWAPFHCFAB : EXTENDED ATTRIBUTE BLOCK
0578 247 SWAPFHCFAB: $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 ALLOCA
05C4 253 AOP=<CBT> : CONTIGUOUS BEST TRY
05E4 254
05E4 255 SWAP_RAB: : OUTPUT RAB
05E4 256 $RAB FAB=SWAPFHCFAB,- : ASSOCIATED FAB
05E4 257 RAC=KEY,- :
05E4 258 RSZ=512,- : RECORD SIZE
05E4 259 RBF=ZEROES : BUFFER ADDRESS
0628 260
53 59 53 2E 0628 261 SWPDEFNAM: .ASCII /.SYS/ : DEFAULT NAME STRING
00000004 062C 262 SWPDEFNAMSZ=-.SWPDEFNAM : SIZE OF DEFAULT NAME STRING
062C 263
062C 264 :
062C 265 : PARSED PARAMETERS FOR SWAP FILE CREATION AND INSTALLATION
062C 266 :
00000000 062C 267 BOO$GL_SIZE: : FILE SIZE
062C 268 .LONG 0 :
00000000 0630 269 BOO$GL_INDEX: : RETURNED PFL INDEX
0630 270 .LONG 0 :
0634 271
0634 272 :
0634 273 : OWN STORAGE:
0634 274 :
0634 275
3A 54 55 50 54 55 4F 0634 276
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$INPUT
063B 291 :
063B 292 : INPUT PARAMETERS:
063B 293 : BUF(AP) -      BUFFER ADDRESS
063B 294 : SIZE(AP)-     MAXIMUM SIZE OF READ
063B 295 :
063B 296 : OUTPUT PARAMETERS:
063B 297 : R0 -          COMPLETION STATUS
063B 298 : R1 -          ACTUAL SIZE OF RECORD IN BYTES
063B 299 :
063B 300 : COMPLETION STATUS CODES:
063B 301 : STANDARD RMS COMPLETION STATUS CODES
063B 302 :
063B 303 :--
00000008 063B 304 SIZE=8 ; OFFSET TO SIZE ARGUMENT
063B 305
00000000 063B 306 .Psect PAGED_CODE rd,nowrt,exe,long
0000 307
0000 308 RIO$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,SY$OPEN ; OPEN THE FILE
01 50 E8 001F 315 BLBS R0,10$ ; BR IF OPEN OK
04 0022 316 RET ; RETURN ERROR STATUS
0023 317 10$: ;
0023 318 $CONNECT (R2) ; CONNECT TO INPUT STREAM
000001F1'EF 23 50 E9 002C 319 BLBC R0,30$ ; BR IF ERROR
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$OUTPUT ONLY)
0053 349 :
0053 350 :
0053 351 : IMPLICIT INPUTS:
0053 352 : INPUT RAB AND FAB
0053 353 :
0053 354 : OUTPUT PARAMETERS:
0053 355 : RO - COMPLETION STATUS CODE FROM RMS
0053 356 :
0053 357 : COMPLETION CODES:
0053 358 : STANDARD RMS COMPLETION STATUS CODES
0053 359 :
0053 360 :--
0053 361 :
00000004 0053 362 BUF=4 ; OFFSET TO BUFFER ADDRESS ARGUMENT
0053 363 :
0053 364 RIO$OUTPUTC:: ; OUTPUT COUNTED STRING
0053 365 .WORD 0 ; ENTRY MASK
51 04 AC DO 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
0063 373 .WORD ^M<R2,R3,R4,R5,R6,R7> ; REGISTER SAVE MASK
52 000002C8'EF 9E 0065 374 MOVAB RIO_OUTTAB,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
45 50 E9 0079 377 $CREATE RIO_OUTFAB ; CREATE TYPING FILE
0086 378 BLBC R0,20$ ; BR IF OPEN FAILED
0089 379 :
00010000 8F CA 0089 380 BICL #BOOCMD$M_TERMINAL,- ; ASSUME NOT A TERMINAL DEVICE
00000000'EF 008F 381 BOO$GL_CMDOPT
00000000'8F E1 0094 382 BBC #DEV$V_TRM,-
00000258'EF 009A 383 RIO_OUTFAB+FAB$SL_DEV,- ; BRANCH IF NOT
0B 009F 384 5$
00010000 8F C8 00A0 385 BISL #BOOCMD$M_TERMINAL,- ; SET AS A TERMINAL DEVICE
00000000'EF 00A6 386 BOO$GL_CMDOPT
00AB 387

```

000001F2'EF	01	90	00AB	388	5\$:	\$CONNECT	(R2)	:	CONNECT TO OUTPUT STREAM
28 A2	04 AC	D0	00B4	389		MOVB	#1,OUTP_OPEN	:	MARK OUTPUT STREAM OPEN
22 A2	08 AC	B0	00BB	390	10\$:	MOVL	BUF(AP),RAB\$L_RBF(R2)	:	SET OUTPUT BUFFER
			00C0	391		MOVW	SIZE(AP),RAB\$Q_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 ASCII
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 ASCII string.
00CF 428 :
00CF 429 : --
00CF 430 BOO$READPROMPT::
00CF 431 .WORD ^M<R2,R3,R4,R5,R6,R7>
04 BC FA00 8F 00 00FC 00CF 432 .LOCC #0,#64000,@PROMPT(AP) : FIND END OF PROMPT STRING
57 51 04 AC C3 00D1 433 .SUBL3 PROMPT(AP),R1,R7 : COMPUTE CHARACTER COUNT
52 000003BC'EF 9E 00DD 434 .MOVAB RIO_INPRAB,R2 : GET INPUT RAB ADDRESS
26 000001F1'EF E8 00E4 435 .BLBS INP_OPEN,15$ : 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,SYSSOPEN : 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 $CONNECT (R2) : CONNECT TO INPUT STREAM
000001F1'EF 42 50 E9 0107 443 .BLBC R0,30$ : BR IF ERROR
20 A2 08 AC B0 0111 444 .MOVW #1,INP_OPEN : MARK INPUT OPEN AND CONNECTED
34 A2 57 90 0116 445 15$:
30 A2 04 AC D0 011C 446 .BEQL NOREAD : SET SIZE FOR INPUT
24 A2 0C AC 01 C1 0121 447 .MOVB R7,RAB$B_PSZ(R2) : IF NO BUFFER THEN NO READ
50 00000000'EF 7D 012A 448 .MOVL PROMPT(AP),RAB$L_PBF(R2) : SET SIZE FOR PROMPT
50 D5 0131 449 .ADDL3 #1,BUF(AP),RAB$L_UBF(R2) : AND PROMPT ADDRESS
2E 19 0133 450 .CLRB @BUF(AP) : AND BUFFER ADDRESS
50 D5 0131 451 .MOVQ BOO$GQ_CMDESC,R0 : INITIALIZE COUNT FOR READ STRING
2E 19 0133 452 .TSTL R0 : GET COMMAND STRING DESCRIPTOR
2E 19 0133 453 .BLSS 50$ : IS THERE ANY COMMAND STRING
: NO, ALREADY USED IT

```

16	14	0135	454	BGTR	40\$	:	YES, USE IT AS RECORD
		0137	455	\$GET	(R2)	:	GET A RECORD
01	50	E8	0140	BLBS	R0,20\$	:	SUCCESS
		04	0143	RET		:	RETURN ERROR STATUS
51	22	A2	3C	0144	458	20\$:	MOVZWL RAB\$W_RSZ(R2),R1
OC	BC	51	33	0148	459		CVTWB R1,@BUF(AP)
			04	014C	460	30\$:	RET
53	OC	AC	DO	014D	461	40\$:	MOVL BUF(AP),R3
	83	50	33	0151	462		CVTWB R0,(R3)+
63	61	50	28	0154	463		MOVCL R0,(R1),(R3)
	50	01	DO	0158	464		MOVL #1,R0
00000000	'EF	01	CE	015B	465		MNEGL #1,BOOS\$GQ_CMDESC
			04	0162	466		RET
50	00000000	'8F	DO	0163	467	50\$:	MOVL #RMS\$_EOF,R0
			04	016A	468		RET
				016B	469	NOREAD:	
52	000002C8	'EF	9E	016B	470		MOVAB RIO_OUTTAB,R2
48	000001F2	'EF	E8	0172	471		BLBS OUTP_OPEN,10\$
	00000273	'EF	94	0179	472		CLRB NAM\$B_ESL+RIO_OUTNAM
				017F	473		\$CREATE RIO_OUTFAB
	44	50	E9	018C	474		BLBC R0,20\$
				018F	475		
00010000	8F	CA	018F	476		BICL	#BOOCMD\$M_TERMINAL,-
00000000	'EF		0195	477			BOOS\$GL_CMDOPT
00000000	'8F	E1	019A	478		BBC	#DEV\$V_TRM,-
00000258	'EF		01A0	479			RIO_OUTFAB+FAB\$L_DEV,-
	OB		01A5	480			5\$
00010000	8F	C8	01A6	481		BISL	#BOOCMD\$M_TERMINAL,-
00000000	'EF		01AC	482			BOOS\$GL_CMDOPT
			01B1	483			
			01B1	484	5\$:	\$CONNECT	(R2)
000001F2	'EF	01	90	01BA	485		MOVAB #1,OUTP_OPEN
28	A2	04	AC	DO	01C1	10\$:	MOVL PROMPT(AP),RAB\$L_RBF(R2)
	22	A2	57	B0	01C6		MOVW R7,RAB\$W_RSZ(R2)
				01CA	488		\$PUT (R2)
			04	01D3	489	20\$:	RET

BOO\$FILOPEN - Routine to open input file

```

01D4 491      .SBTTL BOO$FILOPEN - Routine to open input file
01D4 492      :
01D4 493      : Input Parameters:
01D4 494      : R7 - Address of file name descriptor
01D4 495      : Output Parameters:
01D4 496      : R0 - Completion status
01D4 497      : R8 - RAB address
01D4 498      :
01D4 499      :
58 00000400'EF 9E 01D4 500 BOO$FILOPENW::          : Open file for reads and writes
    16 AB 01 88 01DB 501      MOVAB PRM_INFAB,R8          : Get address of FAB
    1D 11 01DF 502      BISB #FAB$M_PUT,FAB$B_FAC(R8); Set PUT bit
01E1 503      BRB FILOPEN
01E1 504      :
00000430'EF 000001E2'EF 9E 01E1 505 BOO$EXEOPEN::          : Open with default SYSS$SYSTEM:.EXE
00000435'EF 0F 90 01E1 506      MOVAB EXEDEF,PRM_INFAB+FAB$D_DNA
    01EC 507      MOVB #EXEDEF$SIZ,PRM_INFAB+FAB$B_DNS
    01F3 508      :
58 00000400'EF 9E 01F3 509 BOO$FILOPEN::          : Open file for read only
    16 AB 01 8A 01FA 510      MOVAB PRM_INFAB,R8          : Get address of FAB
    8A 01FE 511      BICB #FAB$M_PUT,FAB$B_FAC(R8); Clear PUT bit
    01FE 512      :
    34 AB 67 90 01FE 513 FILOPEN:
    2C AB 04 A7 D0 01FE 514      MOVB (R7),FAB$B_FNS(R8)      : Set size of file name
    0202 515      MOVL 4(R7),FAB$C_FNA(R8)      : and address of name string
    0207 516      $CLOSE FAB=(R8)              : Close if open
    0210 517      $OPEN FAB=(R8)              : Open file for read
    30 AB 000001D7'EF 9E 0219 518      MOVAB PRMDEF,FAB$D_DNA(R8) : Go back to usual default
    35 AB 0B 90 0221 519      MOVB #PRMDEF$SIZ,FAB$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
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::
0259 543      MOVAB PRM_INFAB,R8      : Open file for read only
30 A8 00000400'EF 9E 0259 544      MOVAB EXEDEF,FAB$L_DNA(R8) : Get address of FAB
0259 545      MOVAB #EXEDEF$SIZ,FAB$B_DNS(R8) : Set default name string to
0259 546      MOVAB #EXEDEF$SIZ,FAB$B_DNS(R8) : SYSSYSTEM:.EXE
04 A8 00020000 8F C8 0268 547      BICB #FAB$M_PUT,FAB$B_FAC(R8) : Clear PUT bit
0259 548      BISL #FAB$M_UFO,FAB$L_FOP(R8) : UFO
0259 549      MOVAB (R7),FAB$B_FNS(R8) : Set size of file name
0259 550      MOVL 4(R7),FAB$C_FNA(R8) : and address of name string
0259 551      $CLOSE FAB=(R8) : Close if open
0259 552      $OPEN FAB=(R8) : Open file for read
0259 553      MOVAB PRMDEF,FAB$L_DNA(R8) : Go back to usual default
0259 554      MOVAB #PRMDEF$SIZ,FAB$B_DNS(R8)
0259 555      BLBS R0,10$ : Continue if successful
0259 556      BSBW PUTERROR : Output error message
0259 557      MNEGL #2,R0 : Return failure
04 A8 00020000 8F CA 02A8 558 10$: BICL #FAB$M_UFO,FAB$L_FOP(R8) : Next caller doesn't want UFO
0259 559      MOVZWL FAB$L_STV(R8),R1 : Channel
0259 560      RSB
0259 561

```

```
0286 563 .SBTTL BOO$FILCLOSE - Routine to close currently open file
0286 564 :
0286 565 : Input Parameters:
0286 566 : None
0286 567 : Output Parameters:
0286 568 : R0 - Completion status
0286 569 : R8 - FAB address
0286 570 :
0286 571 :
58 00000400'EF 9E 0286 572 BOO$FILCLOSE:: : Open file for reads and writes
0286 573 MOVAB PRM_INFAB,R8 : Get address of FAB
028D 574 $CLOSE FAB=(R8) : Close if open
05 02C6 575 RSB
02C7 576
```



```

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
21 50 E9 031C 620 $WRITE RAB=(R1) : Do it!
56 0200 C6 9E 0325 621 BLBC R0,20$ : Error
58 D6 0328 622 MOVAB 512(R6),R6 : Advance buffer pointer
D5 59 F5 032D 623 INCL R8 : and block number
032F 624 SOBGTR R9,BOO$WRITEFILE : Repeat
0332 625
51 00000400'EF 9E 0332 626 MOVAB PRM_INFAB,R1 : Get address of FAB
04 50 E9 0339 627 $CLOSE FAB=(R1) : Close file
50 01 3C 0342 628 BLBC R0,20$ : Error
05 0345 629 MOVZWL #1,R0 : Return success
0348 630 RSB
0349 631
50 FCB4' 30 0349 632 20$: BSBW PUTERROR : Output error message
02 CE 034C 633 MNEGL #2,R0 : Set error status
05 034F 634 RSB

```

```

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

```

			03E3	693	\$CREATE	FAB=(R7)	:	Create and open file		
		E8	03EC	694	BLBS	RO,10\$	:	Continue if success		
50	007C80FA	8F	D0	03EF	695	MOVL	#SYSG\$_CREPARFIL,RO	:	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\$_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\$_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	:	:		



```

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$GL_SIZE - SIZE OF FILE TO CREATE IN BLOCKS
045D 736 :
045D 737 : OUTPUT PARAMETERS:
045D 738 : RO - COMPLETION STATUS CODE
045D 739 :--
00FC 045D 740 .ENTRY BOO$CREATE ^M<R2,R3,R4,R5,R6,R7> ; SAVE R2-R7
045F 741
57 00000528'EF DE 045F 742 MOVAL SWAPFHC FAB,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 DO 046B 745 MOVL #<<RMS$_FNM> & ^C<STSSM_SEVERITY>>,RO ; SET WARNING
04 0472 746 RET ; RETURN
0473 747
55 0000062C'EF DO 0473 748 10$: 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$GL_CMOPT,20$ ; 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 DO 0490 753 20$: MOVL R6,SWAPFHCXAB+XAB$_NXT ; SET ADDRESS OF ALLOCATION XAB
00000000'EF 11 E0 0497 754 BBS #BOOCMD$V CONTIG, -
75 049E 755 BOO$GL_CMOPT,CREATE ; BR IF CONTIGUOUS SPECIFIED
049F 756
: Try to open file to see if it exists
049F 757
04 A7 00020000 8F CA 049F 759 BICL #FAB$_UFO,FAB$_FOP(R7); CLEAR USER FILE OPEN
08 A6 20 C8 04A7 760 $OPEN FAB=(R7) ; OPEN
5D 50 E9 04B0 761 BICL #XAB$_CBT,XAB$_AOP(R6); SET FOR CONTIGUOUS BEST TRY ALLOCATION
E9 04B4 762 BLBC RO,CREATE ; IF ERROR, CREATE FILE
04B7 763
: Extend file
04B7 764
51 55 10 A6 C3 04B7 766 SUBL3 XAB$_ALQ(R6),R5,R1 ; DIFF
35 13 04BC 767 BEQL 30$ ; IF THE SAME, THEN NOTHING TO DO
4B 19 04BE 768 BLSS 50$ ; IF LESS THEN NEW FILE
10 A6 51 DO 04C0 769 MOVL R1,XAB$_ALQ(R6)
38 50 E9 04C4 770 $EXTEND FAB=(R7)
04D0 771 BLBC RO,40$
54 000005E4'EF 9E 04D0 773 MOVAB SWAP_RAB,R4 ; RAB ADDRESS
04D7 774 $CONNECT RAB=(R4)
25 50 E9 04E0 775 BLBC RO,40$
38 A4 55 DO 04E3 776 MOVL R5,RAB$_BKT(R4) ; SET LENGTH
04E7 777 $WRITE RAB=(R4) ; FORCE EOF TO SIZE SPECIFIED
15 50 E9 04F0 778 BLBC RO,40$ ; BRANCH ON ERROR
04F3 779 30$: $CLOSE FAB=(R7) ; CLOSE FILE
09 50 E9 04FC 780 BLBC RO,40$ ; EXIT IF ERROR
50 007CA033 8F DO 04FF 781 MOVL #SYSG$_EXTENDED,RO ; SET STATUS
71 11 0506 782 BRB CREATE_SIGNAL ; OK TO SIGNAL
0094 31 0508 783 40$: BRW CREATE_ERROR ; EXIT
050B 784

```

```

04 A7      10 A6  55      DO 050B  785 50$:  $CLOSE FAB=(R7)
          00020000 8F      CB 0514  786
          10 50      E8 0514  787 CREATE: MOVL R5,XAB$ALQ(R6) ; SET ALLOCATION REQUEST SIZE
          7E 50      DD 0518  788          BISL #FAB$M_UFO,FAB$_FOP(R7); SET USER FILE OPEN, SO STV WILL BE CHANNEL
          00000000'GF 07      CB 0520  789          $CREATE FAB=(R7) ; OPEN AND CREATE FILE
          10 50      DD 0529  790          BLBS R0,10$ ; CONTINUE IF NO ERROR
          0C A7      DD 052C  791          PUSHL FAB$_STV(R7) ; SET FAILURE STATUS VALUE
          50 07      CB 052F  792          BICL3 #STSS$M_SEVERITY,R0,-(SP); SET FAILURE STATUS, CONVERTING TO WARNING
          00000000'GF 02      FB 0533  793          CALLS #2,G^LIB$SIGNAL ; SIGNAL THE FAILURE
          60 11      053A  794          BRB CREATE_EXIT ;
          053C  795
          053C  796 10$:  $QIOW_S CHAN=FAB$_STV(R7),- ; ZERO FIRST LONGWORDS IN FILE
          053C  797          FUNC=#IOS_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
          2D 50      E9 0564  801          $DASSGN_S CHAN=FAB$_STV(R7) ; DEASSIGN CHANNEL
          00001070 8F      DO 056F  802          BLBC R0,CREATE_ERROR
          50 07      DO 0572  803          MOVL #SHR$_CREATED,R0 ; RETURN SUCCESS
          0579  804
          0579  805 CREATE_SIGNAL:
          0579  806
          7E 000000B0'EF 9F 0579  807          PUSHAB RSLFNM ; SET FILE NAME ADDRESS
          0000035F'EF 9A 057F  808          MOVZBL RIO_INPNAM+NAM$B_RSL,-(SP) ; SET FILE NAME SIZE
          51 5E      DO 0586  809          MOVL SP,R1 ; GET ADDRESS OF DESCRIPTOR
          51 01      DD 0589  810          PUSHL R1 ; SET FILE NAME DESCRIPTOR ADDRESS
          007C0003 8F      C9 058B  811          PUSHL #1 ; SET FAO COUNT
          7E 50      FB 058D  812          BISL3 #<SYSG$ FACILITY@16!STSSK ;
          00000000'GF 03      FB 0593  813          ,R0,-(SP) ; MAKE SYSGEN MESSAGE
          059C  814          CALLS #3,G^LIB$SIGNAL ; SIGNAL THE ACTION JUST TAKEN
          059C  815
          50 07      DO 059C  816 CREATE_EXIT:
          059C  817          MOVL #1,R0 ; Set sucess
          059F  818
          34 A7      94 059F  819 CREATE_ERROR:
          2C A7      D4 05A2  820          CLRFB FAB$_FNS(R7) ; CLEAR FILE NAME LENGTH
          04 05A5  821          CLRL FAB$_FNA(R7) ; AND FILE NAME ADDRESS
          05A6  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$INSTALL is called to install a second page or swap file. The
05A6 831      :     specified file is opened, the window pointer is removed from the
05A6 832      :     associated channel control block and moved to the page file control
05A6 833      :     block. The PFL address is loaded into the page file control block
05A6 834      :     vector.
05A6 835      :
05A6 836      : Input Parameters:
05A6 837      :
05A6 838      :     SWAPFHCFAB - Specified file name fields filled in.
05A6 839      :     SWAPFHGXAB - File size is stored in XAB
05A6 840      :     BOO$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::
05A6 862      .WORD  ^M<R2,R3,R4,R5,R6,R7>      ; Entry mask
05A8 863      MOVAL  SWAPFHCFAB,R7          ; Get base of FAB
05AF 864      BISL   #FAB$M_UFO,FAB$L_FOP(R7); SET USER FILE OPEN, SO STV WILL BE CHANNEL
05B7 865
05B7 866      10$:  $OPEN  FAB=(R7)          ; Open file for install
05C0 867      BLBS   R0,20$                ; Continue if no error
05C3 868      PUSHAB EXPFNM                 ; Set file name address
05C9 869      MOVZBL RIO_INPNAM+NAM$B_ESL,-(SP) ; Set file name size
05D0 870      MOVL   SP,R1                  ; Get address of descriptor
05D3 871      PUSHL  FAB$L_STV(R7)         ; Set status value
05D6 872      PUSHL  R0                      ; Set status
05D8 873      PUSHL  R1                      ; Set file name descriptor address
05DA 874      PUSHL  #1                      ; Set FAO count
05DC 875      PUSHL  #SHR$_OPENIN!<SYSG$_FACILITY@16> ; Set message status
05E2 876      CALLS  #5,G^CIB$SIGNAL       ; Signal the failure
05E9 877      BRB   30$                    ; Return successfully because
05EB 878      :                                     ; error already signalled
05EB 879
05EB 880      20$:  $CMKRNL_S INITPAGSWPFIL
05FA 881      BLBS   -R0,40$                  ; Branch if success

```

```

04 57 00000528'EF 00FC
A7 00020000 8F C8
      28 50 E8
00000030'EF 9F
7E 00000367'EF 9A
      51 5E DO
      OC A7 DD
      50 DD
      51 DD
      01 DD
007C1098 8F UD
00000000'GF 05 FB
      1B 11
      05EB
      05EB
      OF 50 E8

```

```

00000000'GF 50 DD 05FD 882 25$: PUSHL R0 ; Set failure status
50 0000'8F 01 FB 05FF 883 CALLS #1,G^LIB$SIGNAL ; Signal the failure
3C 0606 884 30$: MOVZWL #$$$_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$INVPGFIL ; Invalidate any saved checkpoints
060C 900 : **JNL** BLBC R0,25$ ; 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$GL_INDEX ; Returned PFL index
7E D4 0614 909 CLRL -(SP) ; Default MAXVBN parameter to default
52 00000528'EF DE 0616 910 MOVAL SWAPFHC$FAB,R2 ; Get base of FAB
52 52 0C A2 CE 061D 911 MNEGL FAB$STV(R2),R2 ; Get channel number and invert
00000000'9F C0 0621 912 ADDL @#CTL$GL_CCBASE,R2 ; Compute base of channel
04 A2 DD 0628 913 PUSHL CCB$WIND(R2) ; Get window pointer address
7E 00000588'EF 01 C3 062B 914 SUBL3 #1,SWAPFHC$XAB+XAB$EBK,-(SP) ; Get size of file
09 00000000'EF 04 E0 0633 915 BBS #BOOCMD$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 DO 064E 923 MOVL CCB$WIND(R2),R0 ; Disconnect file from this process by
00000000'EF 16 0652 924 JSB MMG$RET_BYT_QUOTA ; refund byte quota & clear PID in WCB
52 04 A2 DO 0658 925 MOVL CCB$WIND(R2),R2 ; Get window control block address
0E A2 B6 065C 926 INCW WCB$W-REFCNT(R2) ; Jimmy ref count so WCB sticks around
52 00000528'EF DE 065F 927 MOVAL SWAPFHC$FAB,R2 ; Get base of FAB
0666 928 $DASSGN_S CHAN = FAB$STV(R2) ; Deassign the channel to unlock the file
04 0671 929 30$: RET
0672 930 :
0672 931 :
0672 932 : Clean up after a page file installation has failed
0672 933 :
0672 934 CLNUPPAGFIL:
53 53 0C A7 0038 0672 935 .WORD ^M<R3,R4,R5>
00000000'9F CE 0674 936 MNEGL FAB$STV(R7),R3 ; Get channel number and invert
54 00000630'EF C0 0678 937 ADDL @#CTL$GL_CCBASE,R3 ; Compute base of channel
DO 067F 938 MOVL BOO$GL_INDEX,R4 ; Page file index

```

```

55 00000000'FF44  DO 0686 939      MOVL @MMG$GL_PAGSWPVC[R4],R5 ; Page file control block
    04 A3  OC A5  DO 068E 940      MOVL PFL$SL_WINDOW(R5),CCB$SL_WIND(R3) ; Put window back
                                0693 941      $DASSGN_S_CHAN = FAB$SL_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  DO 06B1 944      MOVL R5,R0 ; Address of PFL block
                51 08 A5 3C 06B4 945      MOVZWL PFL$W_SIZE(R5),R1 ; PFL size
                51 14 A5  CO 06B8 946      ADDL2 PFL$SL_BITMAPSIZ(R5),R1 ; Add the bitmap
                00000000'GF 16 06BC 947      JSB G^EXE$DEANONPGDSIZ ; Deallocate it all
                                06C2 948      SETIPL #0
                                04 06C5 949      RET
0008 06C6 950 ;
0008 06C6 951 ios: .WORD IPL$_SYNCH

```

```

06C8 953      .SBTTL  INSTALL/CREATE ACTION ROUTINES
06C8 954
06C8 955 BOO$SETPGFL::      ; SET PAGEFILE OPTION
00 00000000'EF 04 0000 06C8 956      .WORD  0
E2 06CA 957      BBSS    #BOOCMD$V_PAGEFILE,BOO$GL_CMDOPT,10$
04 06D2 958 10$:      RET
06D3 959
06D3 960 BOO$CRENCONTIG::    ; CREATE NON-CONTIGUOUS FILE
0000 06D3 961      .WORD  0
00000000'EF 20 0000 06D5 962      BISL    #BOOCMD$M_NONCONTIG,BOO$GL_CMDOPT ; SET OPTION BIT
00000000'EF 00020000 8F 04 06DC 963      BICL    #BOOCMD$M_CONTIG,BOO$GL_CMDOPT ; CLEAR OPTION BIT
06E7 964      RET
06E8 965
06E8 966 BOO$CRECONTIG::    ; CREATE CONTIGUOUS FILE
0000 06E8 967      .WORD  0
00000000'EF 20 0000 06EA 968      BICL    #BOOCMD$M_NONCONTIG,BOO$GL_CMDOPT ; CLEAR OPTION BIT
00000000'EF 00020000 8F 04 06F1 969      BISL    #BOOCMD$M_CONTIG,BOO$GL_CMDOPT ; SET OPTION BIT
06FC 970      RET
06FD 971
06FD 972 BOO$FILESIZE::    ; SET FILE SIZE
0000 06FD 973      .WORD  0
0000062C'EF 1C AC 0000 06FF 974      MOVL   TPA$L_NUMBER(AP),BOO$GL_SIZE ;
04 0707 975      RET
0708 976
0708 977 BOO$SETFILNAM::    ;
0000 0708 978      .WORD  0
51 00000528'EF 9E 070A 979      MOVAB  SWAPFHCFAB,R1
34 A1 10 AC 90 0711 980      MOVB   TPA$L_TOKENCNT(AP),FAB$B_FNS(R1) ; SET FILE NAME LENGTH
2C A1 14 AC 04 0716 981      MOVL   TPA$L_TOKENPTR(AP),FAB$L_FNA(R1) ; AND FILE NAME ADDRESS
071B 982      RET
071C 983
071C 984 BOO$NOCHKPNT::    ;
0000 071C 985      .WORD  0
00 00000000'EF 12 0000 071E 986      BBSS    #BOOCMD$V_NOCHKPNT,BOO$GL_CMDOPT,10$
E2 0726 987 10$:      RET
04

```

```

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$GW_OUTLEN:: .WORD 0 ;LENGTH OF STRING
00000100 063E 997 RIO$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$INPUT_FILE, ^M<>
      0729 1037
      14 AC D0 0729 1038      MOVL  TPASL_TOKENPTR(AP),-
00000A11'EF 072C 1039      INNAM_ADDR                ; Save pointer to input file
      10 AC F6 0731 1040      CVTLB TPASL_TOKENCNT(AP),-
00000A10'EF 0734 1041      INNAM_SIZE                ; Save length of file name
00004000 8F C8 0739 1042      BISL  #BOOCMD$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  TPASL_TOKENPTR(AP),-
00000000'EF 074A 1049      BOO$GL_FILEADDR        ; Save pointer to output file
      10 AC F6 074F 1050      CVTLB TPASL_TOKENCNT(AP),-
00000000'EF 0752 1051      BOO$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$DNA ; Set .com default
0000077D'EF 04 90 07D4 1086 MOVB #4,RIO_OUTFAB2+FAB$B_DNS ; Set size of .com
07DB 1087 :
00000000'EF 90 07DB 1088 5$: MOVB BOO$GB_FILELEN,- ; Set up RMS data of size
0000077C'EF 07E1 1089 RIO_OUTFAB2+FAB$B_FNS ; and addr. of file spec.
00000000'EF D0 07E6 1090 MOVL BOO$GL_FILEADDR,- ; Create file
00000774'EF 07EC 1091 RIO_OUTFAB2+FAB$D_FNA ; Branch on error
OD 50 E9 07FE 1093 $CREATE FAB=RIO_OUTFAB2 ; Branch on error
0801 1094 :
0801 1095 7$: $CONNECT RAB = RIO_OUTRAB2 ; and open file
0801 1096 10$: RSB
05 080E 1097 :
080F 1098 :
080F 1099 BOO$OPEN_INPUT_2::
33 00000000'EF OE E1 080F 1100 BBC #BOOCMD$V_INPUT,- ; Branch if no /INPUT
00000A10'EF 90 0811 1101 BOO$GL_CMDOPT,10$ ; Set up RMS data of size
000009A8'EF 081D 1103 MOVB INNAM_SIZE,- ; and addr. of file spec.
00000A11'EF D0 0822 1104 MOVL INNAM_ADDR,- ; and open file
000009A0'EF 0828 1105 RIO_INFAB2+FAB$D_FNA ; branch if error
OD 50 E9 083A 1107 $OPEN FAB=RIO_INFAB2 ; Establish record stream
083D 1108 :
083D 1109 10$: $CONNECT RAB=RIO_INRAB2 ; Establish record stream
05 084A 1110 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
10 00000000'EF E1 084D 1117 BBC #BOOCMD$V SETOUTPUT,-
084F 1118 BOO$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$: MOVAL LIS,-
0000080C'EF 086B 1123 RIO_OUTFAB3+FAB$SL_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$SL_FNA ; and addr. of file spec.
088D 1131
3D 50 E9 088D 1132 $CREATE FAB = RIO_OUTFAB3 ; Create file
089A 1133 BLBC R0,30$ ; 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$V_TRM,-
0000081C'EF 08AE 1138 RIO_OUTFAB3+FAB$SL_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
0B 50 E9 08BF 1143 20$: $CONNECT RAB = RIO_OUTRAB3 ; and open file
08CC 1144 BLBC R0,30$ ; 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 RIOSOutput_line::
08DB 1153
08DB 1154 ; Output Routine: Use special FAB if SET/OUTPUT or /OUTPUT was specified,
08DB 1155 ; RIOSOUTPUT if not.
08DB 1156
25 00000000'EF 0D E1 08DB 1157 BBC #BOOCMD$V OUTPUT,BOO$GL CMDOPT,10$; /OUTPUT = SPECIFIED?
000007BA'EF 0000063C'EF B0 08E3 1158 MOVW RIOS$GW_OUTLEN,RIO_OUTRAB2+RAB$W_RSZ
000007C0'EF 00000646'EF DE 08EE 1159 MOVAL RIOSAB_BUFFER,RIO_OUTRAB2+RAB$L_RBF
3F 11 08F9 1160 $PUT RAB=RIO_OUTRAB2
0906 1161 BRB 30$
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 RIOS$GW_OUTLEN,RIO_OUTRAB3+RAB$W_RSZ
00000854'EF 00000646'EF DE 091B 1165 MOVAL RIOSAB_BUFFER,RIO_OUTRAB3+RAB$L_RBF
0926 1166 $PUT RAB=RIO_OUTRAB3
12 11 0933 1167 BRB 30$
0935 1168
7E 0000063C'EF 3C 0935 1169 20$: MOVZWL RIOS$GW_OUTLEN,-(SP) ; Length
00000646'EF DF 093C 1171 PUSHAL RIOSAB_BUFFER ; Address
F71C CF 02 FB 0942 1172 CALLS #2,RIOSOUTPUT ; Output to SYSSOUTPUT
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
BOOSAB PRMBUF ***** X 02
BOOSA PRMBLK ***** X 03
BOOSCREATE 0000045D RG 03
BOOSCRECONTIG 000006E8 RG 03
BOOSCRENCONTIG 000006D3 RG 03
BOOSEXEOPEN 000001E1 RG 03
BOOSFILCLOSE 000002B6 RG 03
BOOSFILESIZE 000006FD RG 03
BOOSFILOPEN 000001F3 RG 03
BOOSFILOPENW 000001D4 RG 03
BOOSGB_FILELEN ***** X 03
BOOSGL_CMDOPT ***** X 03
BOOSGL_FILEADDR ***** X 03
BOOSGL_INDEX 00000630 R 02
BOOSGL_SIZE 0000062C R 02
BOOSGO_CMDESC ***** X 03
BOOSINITPAGFIL ***** X 03
BOOSINITSWPFIL ***** X 03
BOOSINPUT_FILE 00000727 RG 03
BOOSINSTALL 000005A6 RG 03
BOOSNOCHKPNT 0000071C RG 03
BOOSOPEN_INPUT_2 0000080F RG 03
BOOSOPEN_OUTPUT_2 000007B9 RG 03
BOOSOUTPUT_FILE 00000745 RG 03
BOOSREADFILE 000002C7 RG 03
BOOSREADPROMPT 000000CF RG 03
BOOSRESET_COMMAND 0000079C RG 03
BOOSRESET_IO 00000763 RG 03
BOOSSETFICNAM 00000708 RG 03
BOOSSETPGFL 000006C8 RG 03
BOOSSET_OUTPUT 0000084B RG 03
BOOSUFOOPEN 00000259 RG 03
BOOSWRITEFILE 00000307 RG 03
BOOSWRTFILE 00000350 RG 03
BOOSWRTSYSRFILE 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
CCBSL_WIND = 00000004
CLNUPPAGFIL 00000672 R R 03
COM 00000A15 R R 02
CREATE 00000514 R R 03
CREATE_ERROR 0000059F R R 03
CREATE_EXIT 0000059C R R 03
CREATE_SIGNAL 00000579 R R 03
CTL$GL_CCBASE ***** X 03
DEFNAM 00000000 R X 02
DEV$V_TRM ***** X 03
EXE$DEANONPGDSIZ ***** X 03
EXE$GT_STARTUP ***** X 03
EXEDEF 000001E2 R 02
EXEDEF$SIZ = 0000000F
EXPFNM 00000030 R 02
FAB$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$W_GBC = 00000048
FILNAM$SIZ = 00000080
FILOPEN 000001FE R R 03
INITPAGSWPFIL 0000060C R R 03
INNAM_ADDR 00000A11 R R 02
INNAM_SIZE 00000A10 R R 02
INPNAM 000001C4 R R 02
INPNAM$SZ = 00000009
INP_OPEN 000001F1 R R 02
IOS_WRITEVBLK ***** X 03
IPL$ SYNCH = 00000008
LIB$SIGNAL ***** X 03
LIS 00000A19 R R 02
MMG$GL_NULLPFL ***** X 03

```

MMG\$GL_PAGSWPVC	*****	X	03	RIOSAB_BUFFER	00000646	RG	02
MMG\$RET_BYT_QUOTA	*****	X	03	RIOSAB_INBUFFER	00000874	RG	02
NAM\$B_ESL	= 0000000B			RIOSAB_OUTBUF	0000063E	RG	02
NAM\$B_ESS	= 0000000A			RIOSGL_INBUF_SZ	00000870	RG	02
NAM\$B_NOP	= 00000008			RIOSGW_OUTLEN	0000063C	RG	02
NAM\$B_RSL	= 00000003			RIOSINPUT	00000000	RG	03
NAM\$B_RSS	= 00000002			RIOSOUTPUT	00000063	RG	03
NAM\$C_BID	= 00000002			RIOSOUTPUTC	00000053	RG	03
NAM\$C_BLN	= 00000060			RIOSOUTPUT_LINE	000008DB	RG	03
NAM\$S_ESA	= 0000000C			RIO_INFAB2	00000974	RG	02
NAM\$S_RSA	= 00000004			RIO_INPFAB	0000030C	RG	02
NOREAD	0000016B	R	03	RIO_INPNAM	0000035C	RG	02
OUTFNM	00000634	R	02	RIO_INPRAB	000003BC	RG	02
OUTNAM	00000130	R	02	RIO_INRAB2	000009C4	RG	02
OUTPNAM	000001CD	R	02	RIO_OUTFAB	00000218	RG	02
OUTPNAMSZ	= 0000000A			RIO_OUTFAB2	00000748	RG	02
OUTP_OPEN	000001F2	R	02	RIO_OUTFAB3	000007DC	R	02
OUTSZ	= 00000007			RIO_OUTNAM	00000268	RG	02
PFL\$S_BITMAPSZ	= 00000014			RIO_OUTRAB	000002C8	RG	02
PFL\$S_WINDOW	= 0000000C			RIO_OUTRAB2	00000798	RG	02
PFL\$W_SIZE	= 00000008			RIO_OUTRAB3	0000082C	R	02
PR\$ IPL	*****	X	03	RMS\$ EOF	*****	X	03
PRM\$B_POS	= 00000015			RMS\$ FNM	*****	X	03
PRM\$B_SIZE	= 00000014			RSLFNM	000000B0	R	02
PRM\$C_LENGTH	= 00000032			SHR\$ CREATED	= 00001070		
PRM\$S_ADDR	= 00000000			SHR\$ OPENIN	= 00001098		
PRM\$S_FLAGS	= 00000010			SIZE	= 00000008		
PRM\$T_NAME	= 00000016			SS\$ NORMAL	*****	X	03
PRM\$V_ASCII	= 00000010			ST\$K_INFO	= 00000003		
PRM\$V_NEG	= 0000000C			ST\$M_SEVERITY	= 00000007		
PRMDEF	000001D7	R	02	SWAPALLXAB	000005A4	R	02
PRMDEFSIZ	= 0000000B			SWAPALLXABNC	000005C4	R	02
PRM_INFAB	00000400	R	02	SWAPFHCXAB	00000528	R	02
PRM_INRAB	00000450	R	02	SWAPFHCXAB	00000578	R	02
PRM_OUTFAB	00000494	R	02	SWAP RAB	000005E4	R	02
PRM_OUTRAB	000004E4	R	02	SWPDEFNAM	00000628	R	02
PROMPT	= 00000004			SWPDEFNAMSZ	= 00000004		
PROMPT IN	00000A08	R	02	SYSSCLOSE	*****	GX	03
PUTERROR	*****	X	03	SYSSCMKRN	*****	GX	03
RAB\$B_PSZ	= 00000034			SYSSCONNECT	*****	GX	03
RAB\$B_RAC	= 0000001E			SYSSCREATE	*****	GX	03
RAB\$C_BID	= 00000001			SYSSDASSGN	*****	GX	03
RAB\$C_BLN	= 00000044			SYSS\$EXTEND	*****	GX	03
RAB\$C_KEY	= 00000001			SYSSGET	*****	GX	03
RAB\$C_SEQ	= 00000000			SYSSOPEN	*****	GX	03
RAB\$S_BKT	= 00000038			SYSSPUT	*****	GX	03
RAB\$S_CTX	= 00000018			SYSSQIOW	*****	GX	03
RAB\$S_PBF	= 00000030			SYSSREAD	*****	GX	03
RAB\$S_RBF	= 00000028			SYSSWRITE	*****	GX	03
RAB\$S_ROP	= 00000004			SYSG\$ CREPARFIL	= 007C80FA		
RAB\$S_UBF	= 00000024			SYSG\$ EXTENDED	= 007CA033		
RAB\$M_TPT	= 00000002			SYSG\$ FACILITY	= 0000007C		
RAB\$V_CVT	= 0000001A			SYSG\$ INVPARFIL	= 007C8102		
RAB\$V_PMT	= 0000001E			TPASL_NUMBER	= 0000001C		
RAB\$W_RSZ	= 00000022			TPASL_TOKENCNT	= 00000010		
RAB\$W_USZ	= 00000020			TPASL_TOKENPTR	= 00000014		
RHBUF	000001B0	R	02	WCBSW_REF CNT	= 0000000E		

```

XABSB_AID      = 00000017
XABSB_ALN      = 00000009
XABSB_AOP      = 00000008
XABSB_BKZ      = 00000016
XABSC_ALL      = 00000014
XABSC_ALLLEN   = 00000020
XABSC_FHC      = 0000001D
XABSC_FHCLEN   = 0000002C
XABSC_LBN      = 00000002
XABSL_ALQ      = 00000010
XABSL_EBK      = 00000010
XABSL_LOC      = 0000000C
XABSL_NXT      = 00000004
XABSM_CBT      = 00000020
XABSV_CBT      = 00000005
XABSV_CTG      = 00000007
XABSV_HRD      = 00000000
XABSW_DEQ      = 00000014
XABSW_RF10     = 00000018
XABSW_RF12     = 0000001A
XABSW_RF14     = 0000001C
XABSW_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
\$AB\$\$	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

