

MMM		MMM	AAAAAAAAAA		CCCCCCCCCCCC	RRRRRRRRRRRR		0000000000	
MMM		MMM	AAAAAAAAAA		CCCCCCCCCCCC	RRRRRRRRRRRR		0000000000	
MMM		MMM	AAAAAAAAAA		CCCCCCCCCCCC	RRRRRRRRRRRR		0000000000	
MMMMMMM	MMMMMMM	AAA	AAA	CCC		RRR	RRR	000	000
MMMMMMM	MMMMMMM	AAA	AAA	CCC		RRR	RRR	000	000
MMMMMMM	MMMMMMM	AAA	AAA	CCC		RRR	RRR	000	000
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR		000	000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR		000	000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR		000	000
MMM		MMM	AAAAAAAAAAAAAAAA	CCC		RRR	RRR	000	000
MMM		MMM	AAAAAAAAAAAAAAAA	CCC		RRR	RRR	000	000
MMM		MMM	AAAAAAAAAAAAAAAA	CCC		RRR	RRR	000	000
MMM		MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM		MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM		MMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM		MMM	AAA	AAA	CCCCCCCCCCCC	RRR	RRR	0000000000	
MMM		MMM	AAA	AAA	CCCCCCCCCCCC	RRR	RRR	0000000000	
MMM		MMM	AAA	AAA	CCCCCCCCCCCC	RRR	RRR	0000000000	

32

Sym

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MAC

1998

MAC

MAC

MAC
MAC

333

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MACMAC
MACMAC
MAC

MAC

100

```
PPPPPPPP 222222 DDDDDDDD RRRRRRRR VV VV RRRRRRRR
PPPPPPPP 222222 DDDDDDDD RRRRRRRR VV VV RRRRRRRR
PP PP 22 22 DD DD RR RR VV VV RR RR
PP PP 22 22 DD DD RR RR VV VV RR RR
PP PP 22 22 DD DD RR RR VV VV RR RR
PPPPPPPP 22 22 DD DD RRRRRRRR VV VV RRRRRRRR
PPPPPPPP 22 22 DD DD RRRRRRRR VV VV RRRRRRRR
PP 22 22 DD DD RR RR VV VV RR RR
PP 22 22 DD DD RR RR VV VV RR RR
PP 22 22 DD DD RR RR VV VV RR RR
PP 2222222222 DDDDDDDD RRRRRRRR VV VV RRRRRRRR
PP 2222222222 DDDDDDDD RRRRRRRR VV VV RRRRRRRR
```

```
LL 111111 SSSSSSSS
LL 111111 SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS
```

(2)	60	DECLARATIONS
(3)	84	MAC\$P2DRVR PASS 2 DRIVER
(6)	233	OBJECT FILE OUTPUT ROUTINES
(9)	351	LISTING FILE ROUTINES
(11)	420	MAC\$WRT BLNKLIN WRITE A BLANK LINE TO LISTING
(11)	444	MAC\$WRT\$ST WRITE LINE TO LISTING FILE
(13)	527	MAC\$LST_PAG_HDR WRITE NEW PAGE AND HEADER TO LISTING
(14)	576	MAC\$DEC_OUT_R2L OUTPUT DECIMAL NUMBER TO LISTING
(15)	605	MAC\$DEC_OUT_L2X OUTPUT DECIMAL NUMBER LEFT TO RIGHT
(16)	673	TERMINAL OUTPUT ROUTINES

```
0000 1 .TITLE MACSP2DRV PASS 2 DRIVER MODULE
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35 modules for input to the VAX-11 LINKER.
0000 36
0000 37 ENVIRONMENT: USER MODE
0000 38
0000 39 AUTHOR: Benn Schreiber, CREATION DATE: 20-AUG-78
0000 40
0000 41 MODIFIED BY:
0000 42
0000 43 V03-001 MTR0033 Mike Rhodes 25-Apr-1983
0000 44 Fix link truncation errors.
0000 45
0000 46 V02.08 PCG0002 Peter George 16-Apr-1981
0000 47 Filter out abs psect code.
0000 48
0000 49 V01.07 RN0005 R. Newland 13-Aug-1979
0000 50 Variable symbol storage and remove .ALIGN LONG statements
0000 51
0000 52 V01.07 RN0022 R. Newland 31-Oct-1979
0000 53 Translate SYS$LP_LINES to set lines/page
0000 54
0000 55 V01.06 RN0002 R. Newland 01-Feb-1979
0000 56 Changes for Source Update Merge
0000 57 01 -
```

MACSP2DRVR
V04-000

PASS 2 DRIVER MODULE

I 4

16-SEP-1984 02:12:40 VAX/VMS Macro V04-00
5-SEP-1984 01:49:39 [MACRO.SRC]P2DRVR.MAR;1

Page 2
(1)

0000 58 ;--

```

0000 60      .SBTTL  DECLARATIONS
0000 61      ;
0000 62      ; INCLUDE FILES:
0000 63      ;
0000 64      ;
0000 65      ;
0000 66      ; MACROS:
0000 67      ;
0000 68      ;
0000 69      $RABDEF      ;DEFINE RAB OFFSETS
0000 70      $MAC_GENVALDEF ;DEFINE GENERAL VALUES
0000 71      $MAC_SYMBLKDEF ;DEFINE SYMBOL BLOCK OFFSETS
0000 72      $MAC_CTLFLGDEF ;DEFINE CONTROL FLAG OFFSETS
0000 73      ;
0000 74      ;
0000 75      ; LOCAL DATA
0000 76      ;
0000 77      ;
00000000 78      .PSECT  MAC$RO_DATA,NOEXE,NOWRT,GBL, LONG
0000 79      ;
0000 80      P2$DISPATCH::
0000 81      $MAC_INTCODDEF  DISPATCHTABLE
0000003B 00EC 82      P2$K_MAXCOD=$COUNT      ;MAXIMUM PASS 2 CODE + 1

```

```
00EC 84 .SBTTL MACSP2DRVR PASS 2 DRIVER
00EC 85
00EC 86 :++
00EC 87 : FUNCTIONAL DESCRIPTION:
00EC 88 :
00EC 89 : THIS ROUTINE OUTPUTS THE MODULE HEADER TO THE OBJECT FILE,
00EC 90 : INITIALIZES TO READ THE INTERMEDIATE 'FILE', AND THEN
00EC 91 : PROCESSES THE INTERMEDIATE FILE.
00EC 92 :
00EC 93 :--
00EC 94
00000000 95 .PSECT MAC$RO_CODE_P2,NOWRT,GBL, LONG
0000 96
0000 97 MAC$PASS2 DRVR::
5A 00000000'GF 9E 0000 98 MOVAB G*MAC$AB_OBJBUF,R10 ;R10 POINTS TO OBJECT BUFFER DURING PASS 2
00000000'GF D4 0007 99 CLRL G*MAC$GL_PC ;PC STARTS AT 0
57 D4 000D 100 CLRL R7 ;INITIALIZE STACK POINTER
06 6B 031F 30 000F 101 BSBW MAC$WRT_LST_INI ;INITIALIZE LISTING BUFFER
29 E4 0012 102 BBSC #FLGSV_FIRSTLN,(R11),10$ ;Branch if immediate error
FFE7' 30 0016 103 BSBW MAC$GETLIN ;START WITH FIRST LINE OF FILE
FFE4' 30 0019 104 BSBW MAC$FORM_LINENO ;Form line number and audit trail
00000000'GF 00000000'GF 9E 001C 105 10$: MOVAB G*PSECT$MAIN,G*MAC$GL_PSECTPTR ;START IN BLANK PSECT
0027 107 :
0027 108 : OUTPUT MODULE HEADER INFORMATION
0027 109 :
FFD6' 30 0027 110 BSBW MAC$OBJHDROUT ;OUTPUT MODULE HEADER INFO
002A 111 :
002A 112 : RESET THE INITIAL SETTINGS FOR THE ENABLE/DISABLE AND LIST/NLIST
002A 113 : OPTIONS
002A 114 :
55 00000000'EF 9E 002A 115 MOVAB L*LIST$G_DIRLIST,R5 ;POINT TO THE LIST
05 A5 0B A5 98 0031 116 2$: CVTBL SYM$B_TOKEN(R5),SYM$S_VAL(R5) ;RESET TO THE INITIAL SETTING
55 65 D0 0036 117 MOVL (R5),R5 ;LINK TO NEXT
F6 12 0039 118 BNEQ 2$ ;LOOP FOR ALL
55 00000000'EF 9E 003B 119 MOVAB L*ENB$G_OPTIONS,R5 ;POINT TO ENABLE OPTIONS LIST
05 A5 0B A5 98 0042 120 3$: CVTBL SYM$B_TOKEN(R5),SYM$S_VAL(R5) ;RESET
55 65 D0 0047 121 MOVL (R5),R5
F6 12 004A 122 BNEQ 3$ ;LOOP FOR ALL
004C 123 :
004C 124 : SET UP TO READ INTERMEDIATE 'FILE' FROM VIRTUAL MEMORY
004C 125 :
59 00000000'GF D0 004C 126 MOVL G*MAC$GL_INTQUE,R9 ;POINT TO BEGINNING OF FIRST BLOCK
58 0C A9 9E 0053 127 MOVAB 12(R9),R8 ;R8 POINTS TO FIRST COMMAND
57 D4 0057 128 CLRL R7 ;INIT VALUE STACK POINTER
00000000'GF 5E D0 0059 129 MOVL SP,G*MAC$GL_SAVE_SP ;SAVE STACK POINTER
7E 58 0B A9 C1 0060 130 ADDL3 8(R9),R8,-(SP) ;FIGURE END OF BLOCK ONTO STACK
0065 131 :
0065 132 : AT THIS POINT:
0065 133 :
0065 134 : R10 POINTER INTO OBJECT BUFFER
0065 135 : R9 POINTER TO CURRENT INTERMEDIATE 'FILE' BLOCK
0065 136 : R8 POINTER INTO INTERMEDIATE FILE (INCH-ALONG)
0065 137 : R7 VALUE STACK POINTER
0065 138 :
6E 58 D1 0065 139 PASS_2_LOOP:
0065 140 CMPL R8,(SP) ;ARE WE AT THE END OF A BUFFER?
```

```

        0E 1F 0068 141      BLSSU 10$      ;IF LSSU NO
        59 69 D0 006A 142      MOVL (R9),R9      ;YES--CHAIN TO NEXT BUFFER
        49 13 006D 143      BEQL MAC$EARLY_END ;IF EQL THEN EARLY ENDING
        58 0C A9 9E 006F 144      MOVAB 12(R9),R8 ;POINT TO FIRST ACTION
        6E 58 08 A9 C1 0073 145      ADDL3 8(R9),R8,(SP) ;FIGURE END OF NEW BUFFER
        56 88 9A 0078 146 10$: MOVZBL (R8)+,R6 ;GET LENGTH OF FRAME
        56 02 C2 007B 147      SUBL2 #2,R6 ;COUNT LENGTH AND COMMAND BYTES
        50 88 9A 007E 148      MOVZBL (R8)+,R0 ;GET COMMAND
        05 15 0081 149      BLEQ 20$ ;IF LE THEN SURELY ILLEGAL
        3B 50 91 0083 150      CMPB R0,#P2$K_MAXCOD ;ENSURE LEGAL
        02 19 0086 151      BLSS 30$ ;IF LSS THEN LEGAL
        50 D4 0088 152 20$: CLRL R0 ;0 GOES TO ILLEGAL COMMAND ROUTINE
        008A 153      ;
        008A 154      ; CALL PASS 2 ACTION ROUTINE WITH:
        008A 155      ;
        008A 156      ; R10 POINTER INTO OBJECT BUFFER
        008A 157      ; R9 RESERVED
        008A 158      ; R8 POINTER TO INTERMEDIATE FILE
        008A 159      ; R7 VALUE STACK POINTER
        008A 160      ; R6 LENGTH OF COMMAND IN BYTES
        008A 161      ; R5 POINTER TO TOP OF VALUE STACK (MAC$AL_VALSTACK[R7])
        008A 162      ;
        008A 163      ; REGISTERS R8, R6, R5, AND R4-R0 MAY BE DESTROYED
        008A 164      ;
        FD 8F 56 91 008A 165 30$: CMPB R6,#<-1-2> ;SPECIAL MACRO LINE (-1 FLAG -2 BYTES)?
        06 12 008E 166      BNEQ 40$ ;IF NEQ NO
        56 68 3C 0090 167      MOVZWL (R8),R6 ;YES--GET LENGTH OF TEXT
        56 02 C0 0093 168      ADDL2 #2,R6 ;COUNT THE LENGTH WORD
        7E 58 56 C1 0096 169 40$: ADDL3 R6,R8,-(SP) ;FIGURE START OF NEXT COMMAND
        55 00000000'GF47 DE 009A 170      MOVAL G*MAC$AL_VALSTACK[R7],R5 ;POINT TO TOP OF VALUE STACK
        00000000'EF40 DD 00A2 171      PUSHL L*P2$DISPATCH[R0] ;GET ACTION ROUTINE ADDRESS
        9E 16 00A9 172      JSB @ (SP)+ ;DISPATCH TO ACTION ROUTINE
        58 8E D0 00AB 173      MOVL (SP)+,R8 ;SET POINTER TO NEXT COMMAND
        B5 11 00AE 174      BRB PASS_2_LOOP
        00B0 175      ;
        00B0 176      ; WHEN THE .END DIRECTIVE IS ENCOUNTERED, CONTROL WILL TRANSFER
        00B0 177      ; TO HERE.
        00B0 178      ;
        00B0 179      ; MAC$PASS 2 END::
        5E 00000000'GF D0 00B0 180      MOVL G*MAC$GL_SAVE_SP,SP ;RESET THE STACK POINTER
        05 00B7 181      RSB
        00B8 182      ;
        00B8 183      ;
        00B8 184      ; WE REACHED THE END OF THE INTERMEDIATE FILE BEFORE WE SHOULD HAVE??!!!
        00B8 185      ;
        00B8 186      ; MAC$EARLY END::
        00000000'GF 00 FB 00B8 187      CALLS #0,G*MAC$ERR_INTERN ;REPORT INTERNAL ERROR
        FF3E' 31 00BF 188      BRW P2$END ;AND GO FINISH PASS 2
        00C2 189      ;
        00C2 190      ;
        00C2 191      ; ILLEGAL COMMAND ENCOUNTERED IN INTERMEDIATE BUFFER
        00C2 192      ;
        00C2 193      ; P2$ILG::
        00000000'GF 00 FB 00C2 194      CALLS #0,G*MAC$ERR_INTERN ;REPORT INTERNAL ERROR
        FF34' 31 00C9 195      BRW P2$END ;GO FINISH PASS 2
        00CC 196      ;
        00CC 197      ; STORE PIC CODE COMMAND NOT IMPLEMENTED (THIS SHOULD NEVAH HAPPEN SUH!)
```

MAC\$P2DRVR
V04-000

PASS 2 DRIVER MODULE
MAC\$P2DRVR PASS 2 DRIVER

M 4

16-SEP-1984 02:12:40 VAX/VMS Macro V04-00
5-SEP-1984 01:49:39 [MACRO.SRC]P2DRVR.MAR;1

Page 6
(3)

```
00000000'GF      00      FB      00CC      198      ;  
                  FF2A'  31      00CC      199 P2$SPIC::  
                  00D3      200      CALLS  
                  201      BRW      #0,G^MAC$ERR_INTERN      ;(DO WE REALLY NEED THIS?)  
                                P2$END      ;REPORT INTERNAL ERROR  
                                ;GO FINISH PASS 2
```

```
00D6 203
00D6 204 :++
00D6 205 : FUNCTIONAL DESCRIPTION:
00D6 206 :
00D6 207 :     THIS ROUTINE EMITS A SYMBOL OR PSECT NAME TO THE OBJECT FILE.
00D6 208 :
00D6 209 : INPUTS:
00D6 210 :
00D6 211 :     R6     SYMBOL BLOCK ADDRESS
00D6 212 :
00D6 213 : OUTPUTS:
00D6 214 :
00D6 215 :     SYMBOL NAME AND SIZE OUTPUT TO OBJECT FILE
00D6 216 :
00D6 217 :--
00D6 218
00D6 219 MAC$SYMNAMOUT::
50 0E 6B 2E E0 00D6 220 BBS #FLG$V_DBGOUT,(R11),5$ ; If debugger record
00000000'EF D0 00DA 221 MOVL MAC$GL_PSECTPTR,R0 ; or zero psect
05 13 00E1 222 BEQL 5$ ; Then output record
17 0D A0 03 E1 00E3 223 BBC #PSC$V_REL, - ; Else if abs psect, then
53 04 A6 9A 00E8 224 PSC$W_OPTIONS(R0),20$ ; filter record out
53 56 53 C3 00EC 225 5$: MOVZBL SYM$B_NAME(R6),R3 ; Get offset to symbol count/name
54 83 90 00F0 226 SUBL3 R3,R6,R3 ; and form its address
8A 83 90 00F3 227 MOVB (R3)+,R4 ; Get count and advance pointer to name
FA 54 F5 00F9 228 $OBJ_OUTBYT R4 ;EMIT SIZE OF NAME
05 00FF 230 SOBGIR R4,10$ ;STORE ONE CHARACTER OF NAME
231 20$: RSB ;LOOP FOR WHOLE NAME
```

```
0100 233 .SBTTL OBJECT FILE OUTPUT ROUTINES
0100 234
0100 235 :++
0100 236 : FUNCTIONAL DESCRIPTION:
0100 237 :
0100 238 : OUTPUTS THE BYTE IN R0 TO THE STORE IMMEDIATE COMMAND
0100 239 : ALREADY IN PROGRESS. IF THE FLAG 'FLG$M_STOIMF' IS
0100 240 : FALSE THEN A NEW STORE IMMEDIATE COMMAND IS STARTED WITH
0100 241 : THIS BYTE. THE FIRST BYTE OF A STORE IMMEDIATE COMMAND
0100 242 : IS THE TWO'S COMPLEMENT BYTE COUNT OF THE STRING UP TO
0100 243 : -128. (^0200).
0100 244 :
0100 245 : INPUTS:
0100 246 :
0100 247 : R0 BYTE TO STORE
0100 248 :
0100 249 :--
0100 250
0100 251 MAC$STOIM::
51 0E 6B 2E E0 0100 252 BBS #FLG$V_DBGOUT,(R11),5$ ; If debugger record
00000000'EF D0 0104 253 MOVL MAC$GL_PSECTPTR,R1 ; or zero psect
05 13 010B 254 BEQL 5$ ; Then output record
2D 0D A1 03 E1 010D 255 BBC #PSC$V_REL, - ; Else if abs psect, then
0000'8F 5A B1 0112 256 PSC$W_OPTIONS(R1),30$ ; filter record out
02 1B 0117 257 5$: CMPW R10,#MAC$AB_OBJWRN ;PAST WARNING LIMIT?
59 10 0119 258 BLEQU 10$ ;IF LEQ NO
09 6B 12 E4 011B 259 BSBB MAC$WRTOBJ ;YES--OUTPUT BUFFER
00000000'GF 5A D0 011F 260 10$: BBSC #FLG$V_STOIMF,(R11),20$ ;BRANCH IF STORE IMMEDIATE IN PROGRESS
8A 50 94 0126 261 10$: MOVL R10,G^MAC$GL_STOIMPTR ;NO--START ONE.
80 8F 00000000'FF 97 012B 262 CLRB (R10)+ ;LEAVE ROOM FOR COUNT BYTE AND INIT IT
00000000'FF 91 0128 263 20$: MOVB R0,(R10)+ ;STORE DATA BYTE
04 1B 0131 264 DECB @L^MAC$GL_STOIMPTR ;COUNT THE BYTE
00 6B 12 E2 013B 265 CMPB @L^MAC$GL_STOIMPTR,#^0200 ;DONE 128 YET?
05 013F 266 BLEQU 30$ ;IF LEQU YES
267 BBSS #FLG$V_STOIMF,(R11),30$ ;NO--FLAG STORE IMMEDIATE IN PROGRESS
268 30$: RSB
```

```
0140 270 :++
0140 271 : FUNCTIONAL DESCRIPTION:
0140 272 :
0140 273 : IF THE BUFFER POINTER IS PAST THE WARNING LIMIT, THE OBJECT
0140 274 : BUFFER IS WRITTEN OUT. THE BYTE IN R0 IS THEN STORED IN THE
0140 275 : OBJECT FILE BUFFER.
0140 276 :
0140 277 : INPUTS:
0140 278 :
0140 279 : R0      BYTE TO STORE
0140 280 :
0140 281 :--
0140 282 :
0140 283 MAC$CHKBYT::
0000'8F 5A B1 0140 284 CMPW      R10,#MAC$AB_OBJWRN      ;PAST THE WARNING LIMIT?
02 1B 0145 285 BLEQU     10$                      ;IF LEQU NO
2B 10 0147 286 BSBB      MAC$WRTOBJ          ;YES--OUTPUT RECORD
00 11 0149 287 10$: BRB      MAC$OUTOBJ        ;OUTPUT THE BYTE
014B 288 :AND RETURN
014B 289 :
014B 290 :++
014B 291 : FUNCTIONAL DESCRIPTION:
014B 292 :
014B 293 : OUTPUT THE BYTE IN R0 TO THE OBJECT FILE
014B 294 :
014B 295 : INPUTS:
014B 296 :
014B 297 : R0      BYTE TO OUTPUT
014B 298 :
014B 299 :--
014B 300 :
014B 301 .ENABLE LSB
014B 302 :
014B 303 MAC$OUTOBJ::
51 0E 6B 2E E0 014B 304 BBS      #FLG$V_DBGOUT,(R11),5$ ; If debugger record
00000000'EF D0 014F 305 MOVL     MAC$GL_PSECTPTR,R1 ; or zero psect
05 13 0156 306 BEQL     5$ ; Then output record
16 0D A1 03 E1 0158 307 BBC      #PSC$V_REL, - ; Else if abs psect, then
00000000'GF 00000000'GF 91 015D 308 PSC$W_OPTIONS(R1),20$ ; filter record out
02 13 0168 309 5$: CMPB     G^MAC$GL_RECTYP,G^MAC$AB_OBJBUF ;SAME RECORD TYPE?
08 10 016A 310 BEQL     10$ ;IF EQL YES
016C 311 BSBB      MAC$WRTOBJ          ;NO--WRITE OUT LAST RECORD
016C 312 10$:
016C 313 MAC$OUTOBJ 0::
00 8A 50 90 016C 314 MOVB     R0,(R10)+ ;STORE BYTE IN BUFFER
00 6B 12 E5 016F 315 BBCC     #FLG$V_STOIMF,(R11),20$ ;NO LONGER STORING IMMEDIATE
05 0173 316 20$: RSB
0174 317 :
0174 318 .DISABLE LSB
```

```
0174 320 :++
0174 321 : FUNCTIONAL DESCRIPTION:
0174 322 :
0174 323 : WRITE OUT OBJECT CODE BUFFER TO OBJECT CODE FILE
0174 324 :
0174 325 : INPUTS:
0174 326 :
0174 327 : R10 POINTS TO WHERE WE ARE IN BUFFER
0174 328 :
0174 329 : ***NO REGISTERS USED***
0174 330 :
0174 331 : --
0174 332 :
0174 333 MAC$WRTOBJ::
5A 0001'8F B1 0174 334 CMPW #MAC$AB_OBJBUF+1,R10 : IS BUFFER EMPTY?
37 1E 0179 335 BGEQU 10$ : IF GEQ YES
33 6B 15 E1 017B 336 BBC #FLG$V_OBJXST,(R11),10$ : BRANCH IF NO OBJECT FILE
50 DD 017F 337 PUSHL R0 : NO--SAVE R0
50 5A 00000000'8F C3 0181 338 SUBL3 #MAC$AB_OBJBUF,R10,R0 : FIGURE LENGTH OF RECORD
00000022'EF 50 B0 0189 339 MOVW R0,L^MAC$OBJECT_RAB+RAB$W RSZ : SET RECORD LENGTH IN RAB
0190 340 $PUT RAB=L^MAC$OBJECT_RAB,- : SEND RECORD TO OUTPUT FILE
0190 341 ERR=G^MAC$ERR_PUT : ERROR ROUTINE
03 50 E8 01A3 342 BLBS R0,5$ : BRANCH IF SUCCESSFUL
FE57' 30 01A6 343 BSBW MAC$CLS_DEL_OBJ : CLOSE AND DELETE OBJECT FILE
00000000'GF D6 01A9 344 5$: INCL G^MAC$GL_OBJ_RCNT : COUNT OBJECT RECORD WRITTEN
50 8ED0 01AF 345 POPL R0 : RESTORE R0
5A 00000000'GF 9E 01B2 346 10$: MOVAB G^MAC$AB_OBJBUF,R10 : RESET POINTER INTO BUFFER
8A 00000000'GF 90 01B9 347 MOVB G^MAC$GL_RECTYP,(R10)+ : STORE RECORD TYPE IN BUFFER
00 6B 12 E5 01C0 348 BBCC #FLG$V_STOIMF,(R11),20$ : STOP ANY STORE IMMEDIATE IN PROGRESS
05 01C4 349 20$: RSB
```

```
01C5 351 .SBTTL LISTING FILE ROUTINES
01C5 352
01C5 353 :++
01C5 354 : FUNCTIONAL DESCRIPTION:
01C5 355
01C5 356 : POP THE TOP LONGWORD OFF THE VALUE STACK AND LIST THE NUMBER
01C5 357 : OF BYTES IN R0. UPDATE R5 TO POINT TO THE TOP OF THE VALUE
01C5 358 : STACK AGAIN.
01C5 359
01C5 360 : INPUTS:
01C5 361
01C5 362 : R0 NUMBER OF BYTES TO OUTPUT (1 <= R0 <= 4)
01C5 363 : R5 POINTS TO TOP OF VALUE STACK
01C5 364 : R7 VALUE STACK POINTER
01C5 365
01C5 366 : OUTPUTS:
01C5 367
01C5 368 : R5 POINTS TO NEW TOP OF STACK
01C5 369 : R7 UPDATED
01C5 370 : MAC$GL_PC INCREMENTED BY 1
01C5 371 :
01C5 372 :--
01C5 373
01C5 374 MAC$LIST BYT 0::
55 00000000'GF47 DE 01C5 375 MOVAL G^MAC$AL_VALSTACK[R7],R5 ;POINT R5 TO TOP OF STACK
01CD 376 MAC$LIST BYTES::
50 DD 01CD 377 PUSH R0 ;COPY BYTE COUNT
50 DD 01CF 378 PUSH R0 ;COPY ONCE MORE FOR GOOD LUCK
53 00000000'GF 9E 01D1 379 MOVAB G^MAC$GL_VALUE,R3 ;POINT TO RESULT AREA
63 65 D0 01D8 380 MOVL (R5),(R3) ;COPY VALUE FROM TOP OF STACK
57 D7 01DB 381 DECL R7 ;'POP' THE STACK
75 D5 01DD 382 TSTL -(R5) ;KEEP R5 POINTING TO TOP OF STACK
50 83 9A 01DF 383 10$: MOVZBL (R3)+,R0 ;GET NEXT BYTE
18 10 01E2 384 BSBB MAC$LIST_HEX_BYT ;OUTPUT ONE BYTE
F8 6E F5 01E4 385 SOBGTR (SP),10$ ;LOOP FOR ALL BYTES
8E D5 01E7 386 TSTL (SP)+ ;CLEAN STACK
01E9 387 $INC_PC (SP)+ ;INC. PC AND CLEAN STACK
00000000'GF D7 01EE 388 DECL G^MAC$GL_LIST_PTR ;BACK UP LISTING POINTER
00000000'FF 20 90 01F4 389 MOV B #^A/ /,@[MAC$GL_LIST_PTR ;STORE A SPACE
05 01FB 390 RSB
```

```
01FC 392 :++
01FC 393 : FUNCTIONAL DESCRIPTION:
01FC 394 :
01FC 395 : THIS ROUTINE OUTPUTS ONE BYTE TO THE LISTING FILE IN
01FC 396 : HEX (TWO HEX DIGITS)
01FC 397 :
01FC 398 :--
01FC 399 :
01FC 400 MAC$LST_HEX_BYT::
      50 50 9A 01FC 401 MOVZBL R0,R0 ;ENSURE REST OF WORD ZERO
0001'8F 00000000'GF DD 01FF 402 PUSHL R0 ;SAVE THE BYTE
      15 1A 0201 403 CMPW G^MAC$GL_LIST_PTR,#MAC$AB_LST_OP2+1 ; Time to dump buffer?
0001'8F 0B 6B 27 E0 020A 404 BGTRU 10$ ; No if GTRU
      06 1A 0210 405 BBS #FLG$V UPDFIL,(R11),5$ ; If updated file it is time now
      00000000'GF B1 0219 406 CMPW G^MAC$GL_LIST_PTR,#MAC$AB_LST_END+1 ;TIME TO DUMP THE BUFFER?
      06 1A 021B 407 BGTRU 10$ ;IF GTRU NO
      004B 30 021B 408 5$:
      50 6E D0 021B 409 BSBW MAC$WRTLST
      05 10 0221 410 MOVL (SP),R0 ;REFRESH THE BYTE
      50 8E FC 8F 78 0223 411 10$: BSBB 20$ ;OUTPUT LOW 4 BITS
00000000'GF D7 0228 412 ASHL #-4,(SP)+,R0 ;POSITION TO GET HI 4 BITS
      50 FO 8F 8A 022E 413 20$: DECL G^MAC$GL_LIST_PTR
00000000'FF 0000023E'E0 90 0232 414 BICB2 #^C<^XF>,R0 ;ISOLATE 4 BITS
      05 023D 415 MOVB L^HEX_TAB(R0),@L^MAC$GL_LIST_PTR ;CONVERT AND OUTPUT
      023E 416 RSB
      023E 417
42 41 39 38 37 36 35 34 33 32 31 30 023E 418 HEX_TAB:.ASCII /0123456789ABCDEF/ ;CONVERSION FROM BINARY TO HEX
      46 45 44 43 024A
```

```
024E 420 .SBTTL MAC$WRT_BLNKLN WRITE A BLANK LINE TO LISTING
024E 421
024E 422 :++
024E 423 : FUNCTIONAL DESCRIPTION:
024E 424 :
024E 425 : THIS ROUTINE WRITES A BLANK LINE TO THE LISTING FILE
024E 426 :
024E 427 :--
024E 428
024E 429 MAC$WRT_3_BLNK::
03 DD 024E 430 .PUSHL #3 ;SET TO WRITE 3 BLANK LINES
02 11 0250 431 BRB WRT_BL
0252 432
0252 433 MAC$WRT_2_BLNK::
02 DD 0252 434 .PUSHL #2 ;SET TO WRITE 2 BLANK LINES
06 10 0254 435 WRT_BL: BSBB MAC$WRT_BLNKLN ;WRITE A BLANK LINE
FB 6E F5 0256 436 SOBGR (SP),WRT_BL ;LOOP FOR ALL
8E D5 0259 437 TSTL (SP)+ ;CLEAN STACK
05 025B 438 RSB ;EXIT
025C 439
025C 440 MAC$WRT_BLNKLN::
00000000'GF 00000000'8F D0 025C 441 MOVL #-MAC$K_LIST_SIZE,G^MAC$GL_LINELN ;SET FOR NULL LINE
00 11 0267 442 BRB MAC$WRTST ;WRITE LINE AND RETURN
0269 443
0269 444 .SBTTL MAC$WRTLST WRITE LINE TO LISTING FILE
0269 445
0269 446 :++
0269 447 : FUNCTIONAL DESCRIPTION:
0269 448 :
0269 449 : WRITES THE CURRENT LISTING LINE AND INITIALIZES FOR THE
0269 450 : NEXT LINE.
0269 451 :
0269 452 :--
0269 453
0269 454 MAC$WRTLST::
52 00000000'GF D0 0269 455 MOVL G^MAC$GL_LIST_IT,R2 ;GET CURRENT LISTING FLAG
24 6B 1B E1 0270 456 BBC #FLG$V_MACLTB,(R11),20$ ;BRANCH IF NOT EXPANDING A MACRO
1D 00000005'EF E8 0274 457 BLBS L^LST$G_MACROXPAN+SYM$S_VAL,20$ ;EXPANDING--BRANCH IF LISTING
52 D4 027B 458 CLRL R2 ;CLEAR LISTING FLAG
14 00000005'EF E9 027D 460 BLBC L^LST$G_MACROBIN+SYM$S_VAL,20$ ;BRANCH IF NOT LISTING
0D 6B 0C E4 0284 461 BBSC #FLG$V_MEBLST,(R11),10$ ;SPECIAL LIST FLAG ON?
00000000'GF 00000000'GF D1 0288 463 CMPL G^MAC$GL_PC,G^MAC$GL_SAVE_PC ;SAME PC AS LAST TIME?
03 13 0293 464 BEQL 20$ ;IF EQL YES
52 5E D0 0295 465 10$: MOVL SP,R2 ;NO--SET LISTING FLAG
50 00000000'GF D0 0298 466 20$: MOVL G^MAC$GL_LINELN,R0 ;GET LENGTH OF CURRENT LINE
FFFFFFF'EO 0C 91 029F 467 CMPB #FF,L^MAC$AB_LINEBF-1(R0) ;LAST CHARACTER A FORMFEED?
08 12 02A6 468 BNEQ 30$ ;IF NEQ NO
00000000'GF D7 02A8 469 DECL G^MAC$GL_LINELN ;YES--DON'T COUNT IT
0F 13 02AE 470 BEQL 40$ ;IF EQL ZERO LENGTH LINE
00000000'GF D5 02B0 471 30$: TSTL G^MAC$GL_LIST_LVL ;CHECK THE LISTING LEVEL
07 19 02B6 472 BLSS 40$ ;IF LSS DON'T LIST
09 14 02B8 473 BGTR 50$ ;IF GTR LIST EVERYTHING
01 52 D1 02BA 474 CMPL R2,#1 ;DON'T LIST '.LIST/.NLIST' AT LEVEL 0
07 12 02BD 475 BNEQ 60$ ;IF NEQ NOT .LIST/.NLIST
52 D4 02BF 476 40$: CLRL R2 ;CLEAR LISTING FLAG
```

```

      03 11 02C1 477 BRB 60$
      52 5E D0 02C3 478 50$: MOVL SP,R2 ;SET LISTING FLAG
      52 D5 02C6 479 60$: TSTL R2 ;LISTING FLAG CLEAR?
      56 13 02C8 480 BEQL 90$ ;IF EQL YES--DO NO OUTPUT
      63 6B 09 E1 02CA 481 BBC #FLGSV_LSTXST,(R11),MAC$WRT_LST_INI ;BRANCH IF NO LISTING FILE
      50 DD 02CE 482 PUSHL R0 ;SAVE INDEX INTO LINEBF
      00000000'GF D5 02D0 483 TSTL G^MAC$GL_LINE_CNT ;AT THE END OF A PAGE
      04 15 02D6 484 BLEQ 70$ ;IF LEQ YES--GO DO ONE
      03 6B 0A E1 02D8 485 BBC #FLGSV_NEWPND,(R11),80$ ;BRANCH IF NEW PAGE NOT PENDING
      00B8 30 02DC 486 70$: BSBW MAC$LIST_PAG_HDR ;YES--OUTPUT PAGE HEADER
      50 00000000'GF D0 02DF 487 80$: MOVL G^MAC$GL_LINELN,R0 ;FIGURE SIZE OF LINE
      50 0000'8F A0 02E6 488 ADDW2 #MAC$K_LIST_SIZE,R0 ;FIGURE TOTAL LENGTH OF LINE
      51 00000000'EF 9E 02EB 489 MOVAB L^MAC$LIST_RAB,R1 ;POINT TO LISTING RAB
      22 A1 50 B0 02F2 490 MOVW R0,RAB$W_RSZ(R1) ;STORE RECORD SIZE
      23 6B 09 E1 02F6 491 BBC #FLGSV_LSTXST,(R11),87$ ;BRANCH IF ERROR IN HEADER OUTPUT
      28 A1 00000000'GF 9E 02FA 492 MOVAB G^MAC$AB_LST_END,RAB$L_RBF(R1) ;STORE BUFFER ADDRESS
      0302 493 $PUT RAB=(R1);- ;WRITE THE LINE TO THE LISTING
      0302 494 ERR=G^MAC$ERR_PUT
      03 50 E8 0311 495 BLBS R0,85$ ;BRANCH IF GOOD PUT
      FCE9' 30 0314 496 BSBW MAC$CLOSE_LIST ;CLOSE THE LISTING FILE ON ERROR
      00000000'GF D7 0317 497 85$: DECL G^MAC$GL_LINE_CNT ;DECREMENT LINES LEFT ON PAGE
      50 8ED0 031D 498 87$: POPL R0 ;GET INDEX INTO LINEBF
      FFFFFFFF'EO 0C 91 0320 499 90$: CMPB #FF,L^MAC$AB_LINEBF-1(R0) ;FORM FEED?
      04 12 0327 500 BNEQ 100$ ;IF NEQ NO
      00 6B 0A E3 0329 501 BBBS #FLGSV_NEWPND,(R11),100$ ;YES--FLAG NEW PAGE NEEDED
      00 6B 0C E5 032D 502 100$: BBCC #FLGSV_MEBLST,(R11),110$ ;CLEAR SPECIAL LIST FLAG IF
      0331 503 ;IT GOT SET AND WE WERE NOT
      0331 504 ;IN A MACRO
      0331 505 110$:

```

```
00000000'GF 38 BB 0331 507 MAC$WRT_LST_INIT:
20 6B 00 D4 0331 508 PUSH R3,R4,R5 ;SAVE REGISTERS
0000'8F 2C 0333 509 CLRL G^MAC$GL_LIST_LEN ;ZERO LENGTH OF LINE IN BUFFER
00000000'GF 0339 510 MOVCS #0,(R11) ;FILL LISTING BUFFER WITH SPACES
00000000'GF 0340 511 G^MAC$K_LIST_SIZE,-
00000000'GF 0345 512 G^MAC$AB_LST_END
00000000'GF 9E 0350 513 MOVAB G^MAC$AB_SEQ_NUM,G^MAC$GL_LIST_PTR ;INIT LISTING POINTER
50 02 DD 0356 514 PUSHL G^MAC$GL_VALUE ;SAVE WHAT MIGHT BE HERE
FE5D 30 0362 515 $VPUSH G^MAC$GL_PC ;STACK THE PC
00000000'GF 8ED0 0368 516 MOVZBL #2,R0 ;SET TO LIST TWO BYTES
00000000'GF D0 0365 517 BSBW MAC$LIST_BYT_0 ;LIST THEM
00000000'GF FF 8F 98 0368 518 $DEC_PC #2 ;DON'T INCREMENT PC FOR IT THOUGH
00000000'FF 20 90 036D 519 POPL G^MAC$GL_VALUE ;RESTORE VALUE
00000000'FF 38 BA 0374 520 MOVL G^MAC$GL_PC,G^MAC$GL_SAVE_PC ;SAVE PC
00000000'FF 20 90 037F 521 CVTBL #-1,G^MAC$GL_LIST_IT ;ASSUME LINE IS LISTED
00000000'FF 38 BA 0387 522 DECL G^MAC$GL_LIST_PTR ;BACK UP POINTER
00000000'FF 20 90 038D 523 MOV B A/ /,G^MAC$GL_LIST_PTR ;STORE A SPACE THERE
00000000'FF 38 BA 0394 524 POPR #M<R3,R4,R5> ;RESTORE REGISTERS
00000000'FF 20 90 0396 525 RSB
```

```
0397 527 .SBTTL MAC$LIST_PAG_HDR WRITE NEW PAGE AND HEADER TO LISTING
0397 528
0397 529 :++
0397 530 : FUNCTIONAL DESCRIPTION:
0397 531 :
0397 532 : THIS ROUTINE OUTPUTS A PAGE MARK AND A NEW PAGE HEADER TO
0397 533 : THE LISTING FILE.
0397 534 :
0397 535 :--
0397 536
0397 537 MAC$LIST_PAG_HDR::
0397 538 BBCC #FLG$V NEWPND,(R11),10$ ;CLEAR NEW PAGE PENDING FLAG
0398 539 10$: MOVL G^MAC$GL_LN_PAGE,G^MAC$GL_LINE_CNT ;Reset lines/page
03A6 540 BBS #FLG$V_LSTXST,(R11),20$ ;BRANCH IF LISTING FILE
03AA 541 BRW 50$ ;NO LISTING FILE--GO AWAY
03AD 542 20$: PUSHR #^M<R0,R1,R2,R3,R4,R5> ;SAVE REGISTERS
03AF 543 :***; SASCTIM S TIMBUF=MAC$AL_ATIM_DSC ;GET NEW TIME FOR PAGE HEADER
03AF 544 : PUSHL G^MAC$GL_LIST_PTR ;SAVE CURRENT LISTING POINTER
03B5 545 MOVAB G^MAC$AB_HD_END,G^MAC$GL_LIST_PTR ;POINT TO WHERE PAGE # GOES
03C0 546 INCL G^MAC$GL_LPTAG ;BUMP PAGE COUNTER
03C6 547 MOVL G^MAC$GL_LPTAG,R0 ;GET LISTING PAGE NUMBER
03CD 548 BSBW MAC$DEC OUT_R2L ;OUTPUT TO BUFFER
03D0 549 MOVAB G^MAC$AB_SBT_END-1,G^MAC$GL_LIST_PTR ;POINT TO WHERE SOURCE PAGE # G
03DB 550 MOVAB #^A//,G^MAC$AB_SBT_END-1 ;STORE TERMINAL PAREN
03E2 551 MOVL G^MAC$GL_SRC_PAG,R0 ;GET SOURCE PAGE NUMBER
03E9 552 BSBW MAC$DEC OUT_R2L ;OUTPUT SOURCE PAGE NUMBER
03EC 553 SUBL3 #1,G^MAC$GL_LIST_PTR,R0 ;GET NEXT AVAIL SPOT
03F4 554 MOVAB #^A/(/,(R0) ;STORE OPEN PAREN
03F7 555 MOVAB L^MAC$LIST_RAB,R5 ;POINT TO LISTING RAB
03FE 556 MOVW #MAC$K_HD_SIZE+1,RAB$W_RSZ(R5) ;STORE RECORD SIZE (INCLUDE FORMFEED)
0404 557 MOVAB G^MAC$AB_HD_NEWPG,RAB$C_RBF(R5) ;AND RECORD ADDRESS
040C 558 BSBW 60$ ;WRITE FIRST LINE OF NEW PAGE
040E 559 MOVW #MAC$K_SBT_SIZE,RAB$W_RSZ(R5) ;LENGTH OF SUBTITLE LINE
0414 560 MOVAB G^MAC$AB_SBT_IDNT,RAB$C_RBF(R5) ;AND ADDRESS
041C 561 BSBW 60$ ;WRITE SUBTITLE LINE
041E 562 CLRW RAB$W_RSZ(R5) ;WRITE BLANK LINE
0421 563 BSBW 60$
0423 564 30$: POPL G^MAC$GL_LIST_PTR ;RESTORE LISTING POINTER
042A 565 40$: POPR #^M<R0,R1,R2,R3,R4,R5> ;RESTORE REGISTERS
042C 566 50$: RSB
042D 567 :
042D 568 : WRITE LINE TO LISTING -- R5 POINTS TO RAB
042D 569 :
042D 570 60$: BBC #FLG$V_LSTXST,(R11),70$ ;BRANCH IF LISTING DISABLED
0431 571 $PUT RAB=(R5),ERR=G^MAC$ERR_PUT ;WRITE LINE TO LISTING
0440 572 BLBS R0,70$ ;BRANCH IF GOOD PUT
0443 573 BSBW MAC$CLOSE_LIST ;ELSE CLOSE THE LISTING FILE
0446 574 70$: RSB
```

```
0447 576 .SBTTL MAC$DEC_OUT_R2L OUTPUT DECIMAL NUMBER TO LISTING
0447 577
0447 578 :++
0447 579 : FUNCTIONAL DESCRIPTION:
0447 580 :
0447 581 : THIS ROUTINE OUTPUTS A 5 DIGIT OR LESS DECIMAL NUMBER
0447 582 : TO THE LISTING FILE (IN REVERSE)
0447 583 :
0447 584 : INPUTS:
0447 585 :
0447 586 : R0 NUMBER TO OUTPUT
0447 587 :
0447 588 :--
0447 589
0447 590 MAC$DEC_OUT_R2L::
0447 591 PUSHRR #^M<R3,R4,R5> ;SAVE REGISTERS
0449 592 CMPL R0,#99999 ;CHECK FOR NUMBER TOO LARGE
0450 593 BLEQU 10$ ;IF LEQ NUMBER IS OK
0452 594 MOVL #99999,R0 ;ELSE USE THE MAXIMUM
0459 595 10$: MOVZWL R0,R4 ;SET UP FOR THE EDIV
045C 596 CLRL R5 ;CLEAR HIGH LONGWORD
045E 597 20$: EDIV #10,R4,R4,R0 ;DO A DIVISION BY 10
0463 598 ADDB2 #^A/0/,R0 ;CONVERT DIGIT TO ASCII
0466 599 BSBB MAC$LST_CHAR ;OUTPUT DIGIT TO LISTING BUFFER
0468 600 TSTL R4 ;ARE WE DONE?
046A 601 BNEQ 20$ ;IF NEQ NO
046C 602 POPR #^M<R3,R4,R5> ;YES--RESTORE REGISTERS
046E 603 RSB ;EXIT
```

0001869F 8F 38 BB 0447 591
50 0001869F 8F 50 D1 0449 592
54 50 07 1B 0450 593
50 54 8F D0 0452 594
55 50 3C 0459 595 10\$:
54 54 54 0A 7B 045C 596
50 50 30 80 045E 597 20\$:
4E 10 0466 599
54 D5 0468 600
F2 12 046A 601
38 BA 046C 602
05 046E 603

```
046F 605 .SBTTL MAC$DEC_OUT_L2X OUTPUT DECIMAL NUMBER LEFT TO RIGHT
046F 606
046F 607 :++
046F 608 : FUNCTIONAL DESCRIPTION:
046F 609 :
046F 610 : THIS ROUTINE OUTPUTS A DECIMAL NUMBER LEFT-TO-RIGHT.
046F 611 :
046F 612 : INPUTS:
046F 613 :
046F 614 : R0 NUMBER TO CONVERT
046F 615 : R1 OUTPUT POINTER
046F 616 : OUTPUTS:
046F 617 :
046F 618 : R0 UPDATED OUTPUT POINTER
046F 619 :
046F 620 :--
046F 621
046F 622 MAC$DEC_OUT_L2X::
0078 8F BB 046F 623 PUSH R #^M<R3,R4,R5,R6> ;SAVE REGISTERS
56 51 D0 0473 624 MOVL R1,R6 ;SET OUTPUT POINTER
54 50 D0 0476 625 MOVL R0,R4 ;AND NUMBER TO PRINT
55 D4 0479 626 CLRL R5 ;CLEAR HI WORD
26 10 047B 627 BSBB DEC_OUT ;CONVERT AND OUTPUT THE NUMBER
50 56 D0 047D 628 MOVL R6,R0 ;SET UPDATED POINTER
0078 8F BA 0480 629 POPR #^M<R3,R4,R5,R6> ;RESTORE REGISTERS
05 0484 630 RSB
0485 631
0485 632 :++
0485 633 : FUNCTIONAL DESCRIPTION:
0485 634 :
0485 635 : THIS ROUTINE OUTPUTS A 5 DIGIT DECIMAL NUMBER IN R0 TO
0485 636 : THE LISTING FILE LEFT-TO-RIGHT.
0485 637 :
0485 638 :--
0485 639
0485 640 MAC$DEC_OUT_L2R::
0078 8F BB 0485 641 PUSH R #^M<R3,R4,R5,R6> ;SAVE REGISTERS
54 50 D0 0489 642 MOVL R0,R4 ;COPY THE NUMBER
55 D4 048C 643 CLRL R5 ;CLEAR HIGH WORD
56 00000000'GF D0 048E 644 MOVL G^MAC$GL_LIST_PTR,R6 ;GET THE LISTING POINTER
0C 10 0495 645 BSBB DEC_OUT ;DO THE DIVISION
00000000'GF 56 D0 0497 646 MOVL R6,G^MAC$GL_LIST_PTR ;UPDATE LISTING POINTER
0078 8F BA 049E 647 POPR #^M<R3,R4,R5,R6> ;RESTORE REGISTERS
05 04A2 648 RSB
04A3 649
04A3 650 DEC_OUT:
50 54 54 0A 7B 04A3 651 EDIV #10,R4,R4,R0 ;DIVIDE EM UP
7E 50 81 04A8 652 ADDB3 #^A/0/,R0,-(SP) ;CONVERT REMAINDER TO ASCII AND STACK IT
54 D5 04AC 653 TSTL R4 ;ARE WE DONE?
02 13 04AE 654 BEQL 20$ ;IF EQL YES
F1 10 04B0 655 BSBB DEC_OUT ;NO--RECURSE
86 8E 90 04B2 656 20$: MOVB (SPT)+,(R6)+ ;GET DIGIT BACK AND STORE IT
05 04B5 657 RSB ;RECURSE OR RETURN
04B6 658
04B6 659 :++
04B6 660 : FUNCTIONAL DESCRIPTION:
04B6 661 :
```

MACSP2DRVR
V04-000

PASS 2 DRIVER MODULE
MAC\$DEC_OUT_L2X OUTPUT

M 5

16-SEP-1984 02:12:40 VAX/VMS Macro V04-00
5-SEP-1984 01:49:39 [MACRO.SRC]P2DRVR.MAR;1

Page 19
(15)

```
04B6 662 : WRITE THE CHARACTER IN R0 INTO THE LISTING BUFFER FROM
04B6 663 : RIGHT TO LEFT. THE POINTER INTO THE LISTING BUFFER IS
04B6 664 : DECREMENTED FIRST.
04B6 665 :
04B6 666 :--
04B6 667 :
04B6 668 MAC$LIST_CHAR::
04B6 669 DECL G*MAC$GL_LIST_PTR ;BACK UP THE POINTER
D7 04B6 669 RO,@L*MAC$GL_LIST_PTR ;PUT CHARACTER INTO LINE BUFFER
90 04BC 670 MOVB
05 04C3 671 RSB
```

00000000'GF 50
00000000'FF 50

```
04C4 673 .SBTTL TERMINAL OUTPUT ROUTINES
04C4 674
04C4 675 :++
04C4 676 : FUNCTIONAL DESCRIPTION:
04C4 677 :
04C4 678 : THIS ROUTINE PRINTS THE LINE CONTAINED IN MAC$AB_LINEBF
04C4 679 : AND MAC$AB_LST_END. THE LENGTH OF THE LINE IS CALCULATED
04C4 680 : AND THE LINE IS OUTPUT.
04C4 681 :
04C4 682 :--
04C4 683
04C4 684 MAC$TERM_BLANK::
51 50 D4 04C4 685 CLRL R0 ;ZERO FOR BLANK LINE
51 5E D0 04C6 686 MOVL SP,R1 ;SHOULDN'T MATTER WHERE IT IS
11 11 04C9 687 BRB MAC$PUT_TERM ;GO OUTPUT AND RETURN
04CB 688 MAC$WRITE_TERM::
50 0000'8F 00000000'GF A1 04CB 689 ADDW3 G^MAC$GL_LINELN,#MAC$K_LIST_SIZE,R0 ;COMPUTE SIZE OF LINE
51 00000000'EF 9E 04D5 690 MOVAB L^MAC$AB_LST_END,R1 ;GET ADDRESS OF BUFFER
04DC 691
04DC 692 MAC$PUT_TERM::
52 00000000'EF 9E 04DC 693 PUSH R2 ;SAVE R2
22 A2 50 B0 04DE 694 MOVAB L^MAC$TERM_RAB,R2 ;GET ADDRESS OF TERMINAL RAB
28 A2 51 D0 04E5 695 MOVW R0,RAB$W_RSZ(R2) ;SET THE RECORD SIZE IN THE RAB
04E9 696 MOVL R1,RAB$L_RBF(R2) ;SET THE RECORD ADDRESS
04ED 697 $PUT RAB=(R2),- ;WRITE THE LINE TO THE TERMINAL
04ED 698 ERR=G^MAC$ERR_PUT
52 8ED0 04FC 699 POPL R2 ;RESTORE R2
05 04FF 700 RSB
0500 701
0500 702 .END
```

\$\$TMP1	=	00000002	
\$\$TMP2	=	00000062	
\$COUNT	=	0000003B	
ARG\$K_SIZE	=	000003E8	
AUD\$K_SIZE	=	00000010	
BLNK	=	00000020	
CHRS\$M_COMMA CR	=	00000020	
CHRS\$M_ILL CHR	=	00000040	
CHRS\$M_NUM BER	=	00000010	
CHRS\$M_SPA_MSK	=	00000001	
CHRS\$M_SYM_CH1	=	00000008	
CHRS\$M_SYM_CHR	=	00000004	
CHRS\$M_SYM_DLM	=	00000002	
CHRSV_COMMA CR	=	00000005	
CHRSV_CVTLWC	=	00000061	
CHRSV_ILL CHR	=	00000006	
CHRSV_NOCVT	=	0000007F	
CHRSV_NUM BER	=	00000004	
CHRSV_SPA_MSK	=	00000000	
CHRSV_SYM_CH1	=	00000003	
CHRSV_SYM_CHR	=	00000002	
CHRSV_SYM_DLM	=	00000001	
CR	=	0000000D	
DEC OUT		000004A3	R 04
ENB\$G_OPTIONS		*****	X 04
FF	=	0000000C	
FLG\$M_ALLCHR	=	00000001	
FLG\$M_BOL	=	00000002	
FLG\$M_CHKLPND	=	00100000	
FLG\$M_COMPEXPR	=	00000004	
FLG\$M_CONT	=	00000008	
FLG\$M_CRF	=	40000000	
FLG\$M_CRSEEN	=	00000001	
FLG\$M_DATRPT	=	00000010	
FLG\$M_DBGOUT	=	00004000	
FLG\$M_DLIMSTR	=	00008000	
FLG\$M_ENDMCH	=	00000020	
FLG\$M_EVALEXPR	=	00000040	
FLG\$M_EXPOPT	=	00000080	
FLG\$M_EXTERR	=	00010000	
FLG\$M_EXTWRN	=	00020000	
FLG\$M_FIRSTLN	=	00000200	
FLG\$M_IFSTAT	=	00800000	
FLG\$M_IIF	=	00400000	
FLG\$M_INSERT	=	00000100	
FLG\$M_IRPC	=	20000000	
FLG\$M_LEXOP	=	00000002	
FLG\$M_LSTXST	=	00000200	
FLG\$M_MAC2COL	=	00000800	
FLG\$M_MACL	=	00000800	
FLG\$M_MACLTB	=	08000000	
FLG\$M_MACTXT	=	00010000	
FLG\$M_MEBLST	=	00001000	
FLG\$M_MOREARG	=	00002000	
FLG\$M_MOREINP	=	00000008	
FLG\$M_NEWPND	=	00000400	
FLG\$M_NOREF	=	01000000	

FLG\$M_NTTYPEPC	=	00000020
FLG\$M_NULCHR	=	00040000
FLG\$M_OBJXST	=	00200000
FLG\$M_OPNDCHK	=	00000100
FLG\$M_OPRND	=	00002000
FLG\$M_OPTVFLIDX	=	00001000
FLG\$M_ORDLST	=	00020000
FLG\$M_P2	=	00004000
FLG\$M_RPTIRP	=	10000000
FLG\$M_SEQFIL	=	02000000
FLG\$M_SKAN	=	00008000
FLG\$M_SPECOP	=	00000004
FLG\$M_SPLALL	=	04000000
FLG\$M_STOIMF	=	00040000
FLG\$M_SYM2COL	=	00000400
FLG\$M_TOCLG	=	00080000
FLG\$M_UPAFLG	=	00000010
FLG\$M_UPDFIL	=	00000080
FLG\$M_UPMARG	=	00000040
FLG\$M_XCRF	=	80000000
FLG\$V_ALLCHR	=	00000000
FLG\$V_BOL	=	00000001
FLG\$V_CHKLPND	=	00000014
FLG\$V_COMPEXPR	=	00000002
FLG\$V_CONT	=	00000003
FLG\$V_CRF	=	0000001E
FLG\$V_CRSEEN	=	00000020
FLG\$V_DATRPT	=	00000004
FLG\$V_DBGOUT	=	0000002E
FLG\$V_DLIMSTR	=	0000002F
FLG\$V_ENDMCH	=	00000005
FLG\$V_EVALEXPR	=	00000006
FLG\$V_EXPOPT	=	00000007
FLG\$V_EXTERR	=	00000030
FLG\$V_EXTWRN	=	00000031
FLG\$V_FIRSTLN	=	00000029
FLG\$V_IFSTAT	=	00000017
FLG\$V_IIF	=	00000016
FLG\$V_INSERT	=	00000008
FLG\$V_IRPC	=	0000001D
FLG\$V_LEXOP	=	00000021
FLG\$V_LSTXST	=	00000009
FLG\$V_MAC2COL	=	0000002B
FLG\$V_MACL	=	0000000B
FLG\$V_MACLTB	=	0000001B
FLG\$V_MACTXT	=	00000010
FLG\$V_MEBLST	=	0000000C
FLG\$V_MOREARG	=	0000002D
FLG\$V_MOREINP	=	00000023
FLG\$V_NEWPND	=	0000000A
FLG\$V_NOREF	=	00000018
FLG\$V_NTTYPEPC	=	00000025
FLG\$V_NULCHR	=	00000032
FLG\$V_OBJXST	=	00000015
FLG\$V_OPNDCHK	=	00000028
FLG\$V_OPRND	=	0000000D
FLG\$V_OPTVFLIDX	=	0000002C

MAC\$P2DRVR
Symbol table

PASS 2 DRIVER MODULE

C 6

16-SEP-1984 02:12:40 VAX/VMS Macro V04-00
5-SEP-1984 01:49:39 [MACRO.SRC]P2DRVR.MAR;1

Page 22
(16)

FLGSV_ORDLST	= 00000011		
FLGSV_P2	= 0000000E		
FLGSV_RPTIRP	= 0000001C		
FLGSV_SEQFIL	= 00000019		
FLGSV_SKAN	= 0000000F		
FLGSV_SPECOP	= 00000022		
FLGSV_SPLALL	= 0000001A		
FLGSV_STOIMF	= 00000012		
FLGSV_SYM2COL	= 0000002A		
FLGSV_TOCLG	= 00000013		
FLGSV_UPAFLG	= 00000024		
FLGSV_UPDFIL	= 00000027		
FLGSV_UPMARG	= 00000026		
FLGSV_XCRF	= 0000001F		
HASHSZ	= 0000007F		
HEX TAB	0000023E	R	04
HYPHEN	= 0000002D		
INPSK_BUFSIZ	= 000003E8		
INTSK_BUFSIZ	= 000013F4		
INTSK_BUFWRN	= 00001390		
LSTSG_DIRLIST	*****	X	04
LSTSG_MACROBIN	*****	X	04
LSTSG_MACROXPAN	*****	X	04
LSTSK_BUFSIZ	= 00000086		
LSTSK_L_P PAGE	= 0000003C		
LSTSK_TITLE_SIZ	= 00000028		
MAC\$AB_HD_END	*****	X	04
MAC\$AB_HD_NEWPG	*****	X	04
MAC\$AB_LINEBF	*****	X	04
MAC\$AB_LST_END	*****	X	04
MAC\$AB_LST_OP2	*****	X	04
MAC\$AB_OBJBUF	*****	X	04
MAC\$AB_OBJWRN	*****	X	04
MAC\$AB_SBT_END	*****	X	04
MAC\$AB_SBT_IDNT	*****	X	04
MAC\$AB_SEQ_NUM	*****	X	04
MAC\$AL_VALSTACK	*****	X	04
MAC\$CHRBUT	00000140	RG	04
MAC\$CLOSE_LIST	*****	X	04
MAC\$CLS_DEL_OBJ	*****	X	04
MAC\$DEC_OUT_L2R	00000485	RG	04
MAC\$DEC_OUT_L2X	0000046F	RG	04
MAC\$DEC_OUT_R2L	00000447	RG	04
MAC\$EARY_END	000000B8	RG	04
MAC\$ERR_INTERN	*****	X	04
MAC\$ERR_PUT	*****	X	04
MAC\$FORM_LINENO	*****	X	04
MAC\$GETLIN	*****	X	04
MAC\$GL_INTQUE	*****	X	04
MAC\$GL_LINELN	*****	X	04
MAC\$GL_LINE_CNT	*****	X	04
MAC\$GL_LIST_IT	*****	X	04
MAC\$GL_LIST_LVL	*****	X	04
MAC\$GL_LIST_PTR	*****	X	04
MAC\$GL_LN_PAGE	*****	X	04
MAC\$GL_LPTPAG	*****	X	04
MAC\$GL_OBJ_RCNT	*****	X	04

MAC\$GL_PC	*****	X	04
MAC\$GL_PSECTPTR	*****	X	04
MAC\$GL_RECTYP	*****	X	04
MAC\$GL_SAVE_PC	*****	X	04
MAC\$GL_SAVE_SP	*****	X	04
MAC\$GL_SRCPAGE	*****	X	04
MAC\$GL_STOIMPTR	*****	X	04
MAC\$GL_VALUE	*****	X	04
MAC\$K_HD_SIZE	*****	X	04
MAC\$K_LIST_SIZE	*****	X	04
MAC\$K_SBT_SIZ	*****	X	04
MAC\$LIST_BYTES	000001CD	RG	04
MAC\$LIST_BYT_0	000001C5	RG	04
MAC\$LIST_RAB	*****	X	04
MAC\$LIST_CHAR	000004B6	RG	04
MAC\$LIST_HEX_BYT	000001FC	RG	04
MAC\$LIST_PAGE_HDR	00000397	RG	04
MAC\$OBJECT_RAB	*****	X	04
MAC\$OBJHROUT	*****	X	04
MAC\$OUTOBJ	0000014B	RG	04
MAC\$OUTOBJ_0	0000016C	RG	04
MAC\$PASS2_DRIVER	00000000	RG	04
MAC\$PASS_2_END	000000B0	RG	04
MAC\$PUT_TERM	000004DC	RG	04
MAC\$STOIM	00000100	RG	04
MAC\$SYMNAMOUT	000000D6	RG	04
MAC\$TERM_BLANK	000004C4	RG	04
MAC\$TERM_RAB	*****	X	04
MAC\$WRITE_TERM	000004CB	RG	04
MAC\$WRTLIST	00000269	RG	04
MAC\$WRTOBJ	00000174	RG	04
MAC\$WRT_2_BLNK	00000252	RG	04
MAC\$WRT_3_BLNK	0000024E	RG	04
MAC\$WRT_BLNKLIN	0000025C	RG	04
MAC\$WRT_LST_INI	00000331	RG	04
MAC_SUBSYS	= 0000007D		
OBJ\$K_BUFSIZ	= 00000200		
OPFSM_LASTOPR	= 00002000		
OPFSM_OPTEXP	= 00001000		
OPFSV_LASTOPR	= 0000000D		
OPFSV_OPTEXP	= 0000000C		
P2\$ADD	*****	X	03
P2\$AND	*****	X	03
P2\$ASH	*****	X	03
P2\$ASN	*****	X	03
P2\$AUGPC	*****	X	03
P2\$BDST	*****	X	03
P2\$CHKL	*****	X	03
P2\$DISPATCH	00000000	RG	03
P2\$DIV	*****	X	03
P2\$END	*****	X	03
P2\$EPT	*****	X	03
P2\$ERR	*****	X	03
P2\$ETX	*****	X	03
P2\$FNEWL	*****	X	03
P2\$ILG	000000C2	RG	04
P2\$INFO	*****	X	03

PSC\$M_BYTE	=	00004000
PSC\$M_CON	=	FFFFFFFFFB
PSC\$M_DEFAULT	=	000001C8
PSC\$M_EXE	=	000000C0
PSC\$M_GBL	=	00000010
PSC\$M_LCL	=	FFFFFFFFEF
PSC\$M_LIB	=	00000002
PSC\$M_LONG	=	00004800
PSC\$M_NOEXE	=	FFFFFFFFBF
PSC\$M_NOPIC	=	FFFFFFFFFE
PSC\$M_NORD	=	FFFFFFFF7F
PSC\$M_NOSHR	=	FFFFFFFFDF
PSC\$M_NOVEC	=	FFFFFFFFDF
PSC\$M_NOWRT	=	FFFFFFFFEF
PSC\$M_OVR	=	00000004
PSC\$M_PAGE	=	00006400
PSC\$M_PIC	=	00000001
PSC\$M_QUAD	=	00004C00
PSC\$M_RD	=	00000080
PSC\$M_REL	=	00000008
PSC\$M_SHR	=	00000020
PSC\$M_USR	=	FFFFFFFFFD
PSC\$M_VEC	=	00000200
PSC\$M_WORD	=	00004400
PSC\$M_WRT	=	00000180
PSC\$S_ALIGNMENT	=	00000004
PSC\$V_ALIGNFLG	=	0000000E
PSC\$V_ALIGNMENT	=	0000000A
PSC\$V_EXE	=	00000006
PSC\$V_GBL	=	00000004
PSC\$V_LIB	=	00000001
PSC\$V_OVR	=	00000002
PSC\$V_PIC	=	00000000
PSC\$V_RD	=	00000007
PSC\$V_REL	=	00000003
PSC\$V_SHR	=	00000005
PSC\$V_VEC	=	00000009
PSC\$V_WRT	=	00000008
PSC\$W_FLAG		00000009
PSC\$W_OPTIONS		0000000D
PSECT\$MAIN		*****
RAB\$L_RBF	=	00000028
RAB\$W_RSZ	=	00000022
RDX\$V_BINARY	=	00000000
RDX\$V_DECIMAL	=	00000002
RDX\$V_DOUBLE	=	00000005
RDX\$V_FLOAT	=	00000004
RDX\$V_GFLOAT	=	00000006
RDX\$V_HEX	=	00000003
RDX\$V_HFLOAT	=	00000007
RDX\$V_OCTAL	=	00000001
REG\$PC	=	0000000F
SEMI-	=	0000003B
STB\$K_PG_MISS	=	0000000A
SYM\$B_NAME		00000004
SYM\$B_SEG		0000000C
SYM\$B_TOKEN		0000000B

X 04

SYMSK_BLKSI2	=	0000000D		
SYMSK_MAXLEN	=	0000001F		
SYMSK_TWOCOL	=	00000010		
SYMSL_LINK	=	00000000		
SYMSL_VAL	=	00000005		
SYMSM_ABS	=	00000010		
SYMSM_ASN	=	00000100		
SYMSM_CRFO	=	00002000		
SYMSM_DEBUG	=	00000020		
SYMSM_DEF	=	00000001		
SYMSM_DELMAC	=	00000200		
SYMSM_EPT	=	00000200		
SYMSM_EXTRN	=	00000008		
SYMSM_GLOBL	=	00000004		
SYMSM_LOCAL	=	00000040		
SYMSM_ODBG	=	00000400		
SYMSM_REF	=	00000080		
SYMSM_RELPSECT	=	00000800		
SYMSM_SUPR	=	00004000		
SYMSM_WEAK	=	00000002		
SYMSM_XCRF	=	00001000		
SYMSV_ABS	=	00000004		
SYMSV_ASN	=	00000008		
SYMSV_CRFO	=	0000000D		
SYMSV_DEBUG	=	00000005		
SYMSV_DEF	=	00000000		
SYMSV_DELMAC	=	00000009		
SYMSV_EPT	=	00000009		
SYMSV_EXTRN	=	00000003		
SYMSV_GLOBL	=	00000002		
SYMSV_LOCAL	=	00000006		
SYMSV_ODBG	=	0000000A		
SYMSV_REF	=	00000007		
SYMSV_RELPSECT	=	0000000B		
SYMSV_SUPR	=	0000000E		
SYMSV_WEAK	=	00000001		
SYMSV_XCRF	=	0000000C		
SYMSW_FLAG	=	00000009		
SYSSPOT	=	*****	GX	04
TAB	=	00000009		
WRT_BL	=	00000254	R	04
X1	=	00000033		
X2	=	00080000		

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
. BLANK .	00000000 (0.)	01 (1.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
\$ABSS	00000013 (19.)	02 (2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
MAC\$RO_DATA	000000EC (236.)	03 (3.)	NOPIC	USR	CON	REL	GBL	NOSHR	NOEXE	RD	NOWRT	NOVEC	LONG				
MAC\$RO_CODE_P2	00000500 (1280.)	04 (4.)	NOPIC	USR	CON	REL	GBL	NOSHR	EXE	RD	NOWRT	NOVEC	LONG				

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.04	00:00:01.52
Command processing	104	00:00:00.33	00:00:02.42
Pass 1	227	00:00:03.83	00:00:20.53
Symbol table sort	0	00:00:00.50	00:00:02.03
Pass 2	146	00:00:01.27	00:00:05.01
Symbol table output	44	00:00:00.22	00:00:01.50
Pset synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	554	00:00:06.21	00:00:33.03

The working set limit was 1350 pages.
36602 bytes (72 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 504 non-local and 48 local symbols.
702 source lines were read in Pass 1, producing 27 object records in Pass 2.
15 pages of virtual memory were used to define 13 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	8
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	14

568 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:P2DRVR/OBJ=OBJ\$:P2DRVR MSRC\$:P2DRVR/UPDATE=(ENH\$:P2DRVR)+LIB\$:MACRO/LIB

0227 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

