

MMM	MMM	AAAAAAAAA	CCCCCCCCCCCCC	RRRRRRRRRRRRR	000000000			
MMM	MMM	AAAAAAAAA	CCCCCCCCCCCCC	RRRRRRRRRRRRR	000000000			
MMM	MMM	AAAAAAAAA	CCCCCCCCCCCCC	RRRRRRRRRRRRR	000000000			
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	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	CCC	RRRRRRRRRRRRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRRRRRRRRRRRR	RRR	000	000
MMM	MMM	AAA	AAA	CCC	RRRRRRRRRRRRR	RRR	000	000
MMM	MMM	AAAAAAAAAAAAAAAAA	CCC	RRR	RRR	RRR	000	000
MMM	MMM	AAAAAAAAAAAAAAAAA	CCC	RRR	RRR	RRR	000	000
MMM	MMM	AAAAAAAAAAAAAAAAA	CCC	RRR	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	CCCCCCCCCCCCC	RRR	RRR	000000000	000000000
MMM	MMM	AAA	AAA	CCCCCCCCCCCCC	RRR	RRR	000000000	000000000
MMM	MMM	AAA	AAA	CCCCCCCCCCCCC	RRR	RRR	000000000	000000000

```

AAAAAA      PPPPPPPP      SSSSSSSS      EEEEEEEEEEE      CCCCCCCC      TTTTTTTTTT
AAAAAA      PPPPPPPP      SSSSSSSS      EEEEEEEEEEE      CCCCCCCC      TTTTTTTTTT
AA          AA      PP          PP      SS          EE          CC          TT
AA          AA      PP          PP      SS          EE          CC          TT
AA          AA      PP          PP      SS          EE          CC          TT
AA          AA      PP          PP      SS          EE          CC          TT
AA          AA      PPPPPPPP      SSSSSS      EEEEEEEE      CC          TT
AA          AA      PPPPPPPP      SSSSSS      EEEEEEEE      CC          TT
AAAAAAAAAA      PP          SS          EE          CC          TT
AAAAAAAAAA      PP          SS          EE          CC          TT
AA          AA      PP          SS          EE          CC          TT
AA          AA      PP          SS          EE          CC          TT
AA          AA      PP          SS          EE          CC          TT
AA          AA      PPPSSSSS      EEEEEEEEEEE      CCCCCCCC      TT
AA          AA      SSSSSSSS      EEEEEEEEEEE      CCCCCCCC      TT
...
...
...
...

LL          IIIIIII      SSSSSSSS
LL          IIIIIII      SSSSSSSS
LL          II          SS
LL          II          SS
LL          II          SS
LL          II          SS
LL          II          SSSSSS
LL          II          SSSSSS
LL          II          SS
LL          II          SS
LL          II          SS
LL          II          SS
LLLLLLLLLLLL      IIIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIIII      SSSSSSSS

```

MA Sy SC AL AL AL AR AU BU CH CH CH CH CH CH CH CH CH CH CH CN CR CS EN ER FF FL

(2)	60	DECLARATIONS
(3)	85	PSECT PROCESS .PSECT STATEMENT
(4)	143	SCAN PSECT OPTIONS
(5)	251	PROCESS .SAVE DIRECTIVE
(6)	289	PROCESS .RESTORE DIRECTIVE
(7)	333	ALIGNMENT DIRECTIVE


```
0000 1 .TITLE MAC$APSECT PROCESS PSECT RELATED DIRECTIVES
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 V03.01 RRB030 Rowland R. Bradley 06-Jul-1984
0000 44 Fix broken branch.
0000 45
0000 46 V01.10 RN0023 R. Newland 3-Nov-1979
0000 47 New message codes to get error messages from system
0000 48 message file.
0000 49
0000 50 V01.09 RN0013 R. Newland 27-Sep-1979
0000 51 Use new symbols for PSECT option processing
0000 52
0000 53 V01.08 RN0005 R. Newland 12-Aug-1979
0000 54 Variable symbol names and remove .ALIGN LONG statements
0000 55
0000 56 V01.07 008 B. Schreiber 23-JAN-1979
0000 57 Clear PSECT Block before using it.
```


MAC\$APSECT
V04-000

PROCESS PSECT RELATED DIRECTIVES N 9

16-SEP-1984 02:02:06 VAX/VMS Macro V04-00
5-SEP-1984 01:47:21 [MACRO.SRC]APSECT.MAR;1

Page 2
(1)

0000 58 ;--

MA
VA

Ma
--
-s
-s
TO
50
Th
MA

```

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      $MAC_CTLFLGDEF      ;DEFINE CONTROL FLAGS
0000 70      $MAC_GENVALDEF     ;DEFINE GENERAL VALUES
0000 71      $MAC_INTCODDEF     ;DEFINE INT. BUFFER CODES
0000 72      $MAC_SYMBLKDEF     ;DEFINE SYMBOL/PSECT BLOCK OFFSETS
0000 73      $MACMSGDEF        ; Define message codes
0000 74      :
0000 75      :
0000 76      : EQUATED SYMBOLS:
0000 77      :
0000 78      :
0000 79      :
0000 80      : OWN STORAGE:
0000 81      :
0000 82      :
00000000 83      .PSECT  MAC$RO_CODE_P1,NOWRT,GBL, LONG

```



```
0000 85 .SBTTL PSECT PROCESS .PSECT STATEMENT
0000 86
0000 87 :++
0000 88 : FUNCTIONAL DESCRIPTION:
0000 89 :
0000 90 : THIS ROUTINE FULLY PROCESSES THE .PSECT STATEMENT. THE
0000 91 : OPTIONS ARE PARSED AND A PSECT BLOCK IS ALLOCATED AND
0000 92 : FILLED IN.
0000 93 :
0000 94 :--
0000 95
0000 96 PSECT::
0000 97 BLBS W^ENB$G_LOCALSYMB+SYM$ - VAL,10$ ;BRANCH IF ENABLE LSB
0000 98 BSBW MAC$SET_NEW_LSB ;NO--MAKE A NEW LSB NOW
0000 99 10$: BSBW MAC$SET_PC ;RECORD HIGH PC
0000 100 BSBW MAC$SYM$CNUP ;GET THE PSECT NAME
0000 101 BLBC R0,50$ ;BRANCH IF NO PSECT NAME
0000 102 MOVAB W^MAC$GL_PSC_LIST,R6 ;YES--POINT TO PSECT LIST
0000 103 MOVL R6,R2 ;COPY IN CASE NOTHING IN LIST
0000 104 MOVL SYM$LINK(R6),R5 ;Get next PSECT entry
0000 105 BEQL 20$ ;BRANCH IF NO PSECTS YET
0000 106 BSBW MAC$SRC_LIST ;SEE IF PSECT ALREADY DECLARED
0000 107 MOVL R1,R6 ;COPY RESULT IF FOUND
0000 108 BLBS R0,60$ ;BRANCH IF FOUND
0000 109 20$: PUSHL R2 ;STACK ADDRESS OF PREVIOUS
0000 110 MOVL W^MAC$GL_PSC_BLKP,R0 ;GET POINTER TO PSECT 2K BLOCK
0000 111 BNEQ 40$ ;IF NEQ GO USE IT
0000 112 30$: BSBW MAC$ALL_2_PAGES ;ALLOCATE TWO PAGES
0000 113 MOVL R0,W^MAC$GL_PSC_BLKP ;SAVE ADDRESS FOR LATER
0000 114 MOVZWL #<1024-8>,(R0) ;Set # bytes in first longword
0000 115 MOVAB 8(R0),4(R0) ;SET 2ND LONGWORD AS POINTER TO FREE
0000 116 40$: MOVZBL W^MAC$AB_TMPSYM,R1 ;Get size of name
0000 117 INCL R1 ;Include count byte
0000 118 SUBL2 R1,(R0) ;Subtract from bytes available
0000 119 SUBL2 #PSC$K_BLKSI2,(R0)+ ;and subtract fixed part size
0000 120 BLSS 30$ ;IF LSS NO--GO ALLOCATE ANOTHER
0000 121 MOVL (R0),R6 ;GET POINTER TO PSECT BLOCK
0000 122 ADDL2 R1,(R0) ;Set pointer to end of allocated
0000 123 ADDL2 #PSC$K_BLKSI2,(R0) ;block
0000 124 MOVAB R1,-(SP) ;Save total length
0000 125 MOVAB R1,W^MAC$AB_TMPSYM,(R6) ;Copy symbol count/name
0000 126 MOVL R3,R6 ;Save pointer to block
0000 127 MOVAB #0,(SP),#0,#PSC$K_BLKSI2,(R3) ;Clear rest of block
0000 128 MOVAB (SP)+,SYM$B_NAME(R6) ;Store offset to name
0000 129 POPL R2 ;GET POINTER TO PREVIOUS PSECT BLOCK
0000 130 MOVL SYM$LINK(R2), - ;Link in new PSECT block
0000 131 SYM$LINK(R6)
0000 132 MOVL R6,SYM$LINK(R2) ; ...
0000 133 BRB 60$
0000 134 50$: MOVAB W^PSECT$BLANK,R6 ;USE THE BLANK PSECT
0000 135 60$: BSBW2 #SYM$M_REF,PSC$W_FLAG(R6) ;FLAG PSECT AS REFERENCED
0000 136 BSBW MAC$PSC_OPT_SCN ;SCAN PSECT OPTIONS
0000 137 $INTOUT LW INT$-PSECT,R6 ;SWITCH TO NEW PSECT IN PASS 2
0000 138 MOVL PSC$LINK_MAXLGTH(R6),W^MAC$GL_PC ;SET NEW PC
0000 139 MOVZBL PSC$B_SEG(R6),W^MAC$GL_PSECT ;AND NEW PSECT
0000 140 MOVL R6,W^MAC$GL_PSECTPTR
0000 141 RSB
```

03 0005'CF E8 0000 97
FFF8' 30 0005 98
FFF5' 30 0008 99
FFF2' 30 000B 100
6A 50 E9 000E 101
56 0000'CF 9E 0011 102
52 56 D0 0016 103
55 66 D0 0019 104
09 13 001C 105
FFDF' 30 001E 106
56 51 D0 0021 107
59 50 E8 0024 108
52 DD 0027 109
50 0000'CF D0 0029 110
12 12 002E 111
FFCD' 30 0030 112
0000'CF 50 D0 0033 113
60 03F8 8F 3C 0038 114
04 A0 08 A0 9E 003D 115
51 0000'CF 9A 0042 116
51 D6 0047 117
60 51 C2 0049 118
80 13 C2 004C 119
DF 19 004F 120
56 60 D0 0051 121
60 51 C0 0054 122
60 13 C0 0057 123
7E 51 90 005A 124
66 0000'CF 51 28 005D 125
56 53 D0 0063 126
63 13 00 6E 00 2C 0066 127
04 A6 8E 90 006C 128
52 8ED0 0070 129
66 62 D0 0073 130
0076 131
62 56 D0 0076 132
05 11 0079 133
56 0000'CF 9E 007B 134
09 A6 0080 8F A8 0080 135
001A 30 0086 136
0089 137
0000'CF 05 A6 D0 0091 138
0000'CF 0C A6 9A 0097 139
0000'CF 56 D0 009D 140
05 00A2 141


```
00A3 143 .SBTTL SCAN PSECT OPTIONS
00A3 144
00A3 145 MAC$PSC_OPT SCN:
OC 09 A6 57 DD 00A3 146 PUSH R7 ;SAVE R7
57 0D A6 00 E1 00A5 147 BBC #SYM$V_DEF,SYM$W_FLAG(R6),10$ ;BRANCH IF PSECT NOT DEFINED
7E 57 3C 00AA 148 MOVZWL PSC$W_OPTIONS(R6),R7 ;DEFINED---GET DEFINED OPTIONS
7E 57 D0 00AE 149 MOVL R7,-(SP) ;COPY TO -(SP) FOR POSITIVE OPTIONS
09 11 00B1 150 MCOML R7,-(SP) ;COPY COMPLEMENTED OPTIONS TO -(SP)
57 01C8 8F 3C 00B4 151 BRB 20$
7E D4 00B6 152 10$: MOVZWL #PSC$M_DEFAULT,R7 ;NEW PSECT--SET DEFAULT OPTIONS
7E D4 00BB 153 CLRL -(SP) ;CLEAR POS. AND NEG. OPTIONS
7E D4 00BD 154 CLRL -(SP) ;...
00BF 155 20$:
00BF 156
00BF 157 OPTION_SCAN:
OD 5A 91 00BF 158 CMPB R10,#CR ;END OF LINE?
03 12 00C2 159 BNEQ 10$ ;IF NEQ NO
00AF 31 00C4 160 5$: BRW 130$ ;YES--EXIT
FF36' 30 00C7 161 10$: BSBW MAC$SKIPSP ;SKIP SPACES
2C 5A 91 00CA 162 CMPB R10,#^A/,/ ;SCAN TO A COMMA?
08 12 00CD 163 BNEQ 20$ ;IF NEQ NO
FF2E' 30 00CF 164 BSBW MAC$GETCHR ;YES--SKIP IT
FF2B' 30 00D2 165 BSBW MAC$SKIPSP ;SKIP SPACES AGAIN
E8 11 00D5 166 BRB OPTION_SCAN ;CK FOR EOL AGAIN
OD 5A 91 00D7 167 20$: CMPB R10,#CR ;GET TO EOL?
E8 13 00DA 168 BEQL 5$ ;IF EQL YES--ALL DONE
04 E1 00DC 169 25$: BBC #CHR$V_NUM BER,- ;BRANCH IF NOT NUMERIC
34 0000'CA 00DE 170 W^MAC$AB_CMSK TAB(R10),60$
08 6E 0E E1 00E2 171 BBC #PSC$V_ALIGNFLG,(SP),30$ ;YES--WAS ALIGNMENT SEEN?
00CC 30 00E6 172 BSBW OPTION_CONFLICT ;YES--REPORT CONFLICT
FF14' 30 00E9 173 BSBW MAC$SKP_OPR ;SKIP TO COMMA OR EOL
D1 11 00EC 174 BRB OPTION_SCAN ;AND SCAN NEXT OPTION
FF0F' 30 00EE 175 30$: BSBW MAC$DNUMBER ;ACCUMULATE NUMERIC ALIGNMENT
55 0000'CF D0 00F1 176 MOVL W^MAC$GL_VALUE,R5 ;GET ALIGNMENT SCANNED
09 55 D1 00F6 177 CMPL R5,#9 ;LEGAL ALIGNMENT?
0B 1B 00F9 178 BLEQU 40$ ;IF LEQU YES
55 09 9A 00FB 179 MOVZBL #9,R5 ;NO--USE PAGE ALIGNMENT
00FE 180 $MAC_ERR INVALIDN ;Get error message code
55 FEFA' 30 0103 181 BSBW MAC$ERRORLN ;REPORT ERROR TO PASS 2
55 55 0A 78 0106 182 40$: ASHL #PSC$V_ALIGNMENT,R5,R5 ;Position alignment
00 55 0E E3 010A 183 BBCS #PSC$V_ALIGNFLG,R5,50$ ;SET ALIGNMENT FLAG
57 55 C8 010E 184 50$: BISL2 R5,R7 ;SET IN OPTIONS
6E 55 C8 0111 185 BISL2 R5,(SP) ;SET IN COMPL. OPTIONS FOR CHECK
A9 11 0114 186 BRB OPTION_SCAN ;CONTINUE SCANNING
0116 187
0116 188 ; NOT A NUMBER
0116 189
FEE7' 30 0116 190 60$: BSBW MAC$SYMSCNUP ;SCAN THE OPTION NAME
OD 50 E8 0119 191 BLBS R0,70$ ;BRANCH IF OPTION SCANNED
011C 192 $MAC_ERR DIRSYN ;No--get message code
FEDC' 30 0121 193 BSBW MAC$ERRORLN ;REPORT ERROR TO PASS 2
FED9' 30 0124 194 BSBW MAC$SKP_OPR ;SKIP TO COMMA OR EOL
96 11 0127 195 BRB OPTION_SCAN ;SCAN NEXT OPTION
0129 196
0129 197 ; LOOK UP THE OPTION
0129 198
55 0000'CF 9E 0129 199 70$: MOVAB W^PSC$G_OPTIONS,R5 ;POINT TO THE OPTIONS
```



```

      FECF' 30 012E 200      BSBW  MAC$SRC_LIST      ;LOOK IT UP
      OB 50  E8 0131 201      BLBS  R0,80$          ;BRANCH IF FOUND
      FE4'   30 0134 202      $MAC_ERR NOTPSECOPT    ; No--get error code
      FF80   31 0139 203      BSBW  MAC$ERRORLN      ;ISSUE ERROR TO PASS 2
      55 05 A1 32 013C 204      BRW   OPTION_SCAN     ;SCAN NEXT OPTION
      55 16 14 0143 205 80$:  CVTWL  SYMSL_VAL(R1),R5 ;GET BITS FOR OPTION
      55 55 D2 0145 206      BGTR  100$            ;IF GTR NOT COMPLEMENT OF OPTION
      04 AE 55 D3 0148 207      MCOML R5,R5          ;COMPLEMENTED--GET UNCOMPLEMENTED
      05 13 014C 208      BITL  R5,4(SP)          ;POSITIVE SET?
      0064 30 014E 209      BEQL  90$              ;IF EQL NO
      20 11 0151 210      BSBW  OPTION_CONFLICT     ;YES--ISSUE MESSAGE
      57 55 CA 0153 211      BRB   120$            ;CONTINUE SCAN
      6E 55 C8 0156 212 90$:  BICL2 R5,R7          ;CLEAR OPTION
      18 11 0159 213      BISL2 R5,(SP)           ;SET IN NEGATIVE MASK
      015B 214      BRB   120$                    ;CONTINUE SCANNING
      015B 215      ;
      015B 216      ; POSITIVE SENSE OPTION
      015B 217      ;
      6E 55 D3 015B 218 100$:  BITL  R5,(SP)        ;WAS COMPLEMENT SET?
      05 13 015E 219      BEQL  110$              ;IF EQL NO
      0052 30 0160 220      BSBW  OPTION_CONFLICT     ;YES--REPORT ERROR
      0E 11 0163 221      BRB   120$              ;CONTINUE SCANNING
      57 55 C8 0165 222 110$:  BISL2 R5,R7          ;SET IN OPTIONS WORD
      04 AE 55 C8 0168 223      BISL2 R5,4(SP)       ;AND IN POSITIVE MASK
      03 55 0E E1 016C 224      BBC   #PSC$V_ALIGNFLG,R5,120$ ;IS THIS AN ALIGNMENT OPTION?
      6E 55 C8 0170 225      BISL2 R5,(SP)          ;YES--SET IN NEG. MASK
      FF49 31 0173 226 120$:  BRW   OPTION_SCAN     ;CONTINUE SCANNING OPTIONS
      0176 227      ;
      0176 228      ; DONE SCANNING OPTIONS
      0176 229      ;
      33 09 A6 00 E0 0176 230 130$:  BBS   #SYMS$V_DEF,PSC$W_FLAG(R6),150$ ;BRANCH IF NOT NEW PSECT
      09 A6 0081 8F A8 017B 231      BISW2 #SYMS$M_DEF!SYMS$M_REF,PSC$W_FLAG(R6) ;NEW--MARK DEFINED
      0181 232      ;AND REFERENCED
      0D A6 57 B0 0181 233      MOVW  R7,PSC$W_OPTIONS(R6) ;SET THE OPTIONS
      0185 234      $INTOUT_LW INT$ NEWP,R6          ;DEFINE NEW PSECT IN PASS 2
      0000'CF D6 018D 235      INCL  W*MAC$GL_PSC_MAX ;COUNT ANOTHER PSECT
      0100 8F 0000'CF B1 0191 236      CMPW  W*MAC$GL_PSC_MAX,#256 ;DO WE HAVE TOO MANY PSECTS?
      08 1B 0198 237      BLEQU  140$              ;IF LEQU NO
      FE5E' 30 019A 238      $MAC_ERR TOOMNYPSECT    ; Yes--get code
      0C A6 0000'CF 90 01A2 239      BSBW  MAC$ERRORLN ;SEND TO PASS 2
      05 A6 D4 01A8 240 140$:  MOVB  W*MAC$GL_PSC_MAX,PSC$B_SEG(R6) ;SET THE PSECT NUMBER
      0F A6 D4 01AB 241      CLRL  PSC$L_MAXLGTH(R6) ;CLEAR MAX LENGTH
      5E 08 C0 01AE 242      CLRL  PSC$L_CURLOC(R6) ;START AT 0
      57 8ED0 01B1 243 150$:  ADDL2 #2*4,SP        ;CLEAR TWO WORDS FROM STACK
      05 01B4 244      POPL  R7                    ;RESTORE R7
      01B5 245      RSB
      01B5 246      ;
      01B5 247      OPTION_CONFLICT:
      FE43' 31 01BA 248      $MAC_ERR PSECOPCNFL    ; Get message code
      01BA 249      BRW   W*MAC$ERRORLN          ;ISSUE MESSAGE AND RETURN
```



```
01BD 251 .SBTTL PROCESS .SAVE DIRECTIVE
01BD 252
01BD 253 ;++
01BD 254 ; FUNCTIONAL DESCRIPTION:
01BD 255 ;
01BD 256 ; CSAVE IS CALLED TO PROCESS THE .SAVE DIRECTIVE. THE
01BD 257 ; NUMBER OF THE CURRENT PSECT IS SAVED IN THE PSECT
01BD 258 ; SAVE BUFFER. THE PC FOR THE CURRENT PSECT IS SAVED
01BD 259 ; IN PSC$$_CURLOC OF THE CURRENT PSECT'S PSECT SYMBOL BLOCK.
01BD 260 ;
01BD 261 ;--
01BD 262
01BD 263 CSAVE::
01BD 264 MOVZBL W^MAC$GL_PSC_SBP,R6 ;DIRECTIVE = KSAVE
01BD 265 CMPB R6,#31 ;GET BUFFER INDEX
01BD 266 BLEQU 10$ ;OVERFLOW?
01BD 267 $MAC_ERR PSECBUFVF ;IF LEQU NO
01BD 268 MOVL #CR,R10 ;Yes--get message code
01BD 269 BRW MAC$ERRORLN ;FORCE END OF LINE
01BD 270 10$: MOVB W^MAC$GL_PSECT,L^MAC$AB_PSC_SBF(R6) ;ISSUE AND RETURN
01BD 271 INCL W^MAC$GL_PSC_SBP ;SAVE PSECT NUMBER
01BD 272 MOVL W^MAC$GL_PSECTPTR,R2 ;BUMP THE INDEX
01BD 273 MOVL W^MAC$GL_PC,PSC$$_CURLOC(R2) ;GET POINTER TO PSECT BLOCK
01BD 274 $INTOUT_LW INT$ SAVE,R2 ;SAVE CURRENT PC
01BD 275 BSBW MAC$SYMSCNUP ;ISSUE SAVE TO PASS 2
01BD 276 BLBS R0,30$ ;SEE IF THERE IS AN ARGUMENT
01BD 277 CMPB R10,#CR ;BRANCH IF THERE WAS
01BD 278 BEQL 40$ ;NO--BUT DID WE GET TO EOL?
01BD 279 20$: $MAC_ERR DIRSYNX ;IF EQL YES
01BD 280 BRW MAC$ERRORLN ;No--call it directive syntax
01BD 281 30$: MOVAB W^MAC$GL_L$BNAM,R5 ;REPORT THE ERROR
01BD 282 BSBW MAC$SRC_LIST ;POINT TO LSB NAMES
01BD 283 BLBC R0,20$ ;SEE IF IT IS LEGAL FOR LSB
01BD 284 MOVL W^MAC$GL_L$B,L^MAC$AL_PSC_SLB[R6] ;BRANCH IF ILL ARG
01BD 285 BRB 50$ ;YES--SAVE LSB
01BD 286 40$: CLRL L^MAC$AL_PSC_SLB[R6] ;AND EXIT
01BD 287 50$: RSB ;ENSURE NO LSB SAVED
```

```
0224 289 .SBTTL PROCESS .RESTORE DIRECTIVE
0224 290
0224 291 :++
0224 292 : FUNCTIONAL DESCRIPTION:
0224 293 :
0224 294 : CRESTO IS CALLED TO PROCESS THE .RESTORE PSECT DIRECTIVE.
0224 295 : CODE IS EMITTED TO PASS 2 TO SWITCH PSECTS.
0224 296 :
0224 297 :--
0224 298
0224 299 CRESTO::
0224 300 BSBW MAC$SET_PC ;DIRECTIVE = KRESTORE
0224 301 MOVL W^MAC$GC_PSC_SBP,R6 ;RECORD HIGH PC
0224 302 BGTR 10$ ;GET SAVE BUFFER INDEX
0224 303 $MAC_ERR PSECBUFUND ;IF GTR OK
0224 304 BSBW MAC$ERRORPT ;No--get error code
0224 305 RSB ;REPORT TO PASS 2
0224 306 10$: DECL R6 ;RETURN
0224 307 MOVL R6,W^MAC$GL_PSC_SBP ;BACK UP THE INDEX
0224 308 MOVZBL L^MAC$AB_PSC_SBF(R6),R0 ;SAVE INDEX
0224 309 CMPB R0,#1 ;GET THE PSECT INDEX
0224 310 BNEQ 20$ ;BLANK PSECT?
0224 311 MOVAB W^PSECT$BLANK,R5 ;IF NEQ NO
0224 312 BRB 50$ ;YES--SET FOR IT
0224 313 20$: MOVL W^MAC$GL_PSC_LIST,R5 ;POINT TO PSECT LIST
0224 314 BEQL MAC$PSC_RES_ERR ;IF EQL NOTHING THERE
0224 315 30$: CMPB R0,PSC$B_SEG(R5) ;IS THIS THE RIGHT PSECT?
0224 316 BEQL 50$ ;IF EQL YES
0224 317 MOVL (R5),R5 ;NO--LINK TO NEXT
0224 318 BNEQ 30$ ;IF NEQ KEEP LOOKING
0224 319 BRB MAC$PSC_RES_ERR ;PSECT RESTORE ERROR
0224 320 50$: $INTOUT LW INT$-REST,R5 ;ISSUE RESTORE TO PASS 2
0224 321 60$: MOVL R5,W^MAC$GL_PSECTPTR ;SET NEW PSECT POINTER
0224 322 MOVZBL PSC$B_SEG(R5),W^MAC$GL_PSECT ;AND NEW PSECT NUMBER
0224 323 MOVL PSC$L_CURLOC(R5),W^MAC$GL_PC ;SET NEW PC
0224 324 MOVL L^MAC$AL_PSC_SLB[R6],R0 ;GET THE SAVED LSB NUMBER
0224 325 BEQL 70$ ;IF EQL NO SAVED PSECT NUMBER
0224 326 MOVL R0,W^MAC$GL_LSB ;SET NEW LSB
0224 327 70$: RSB
0224 328
0224 329 MAC$PSC_RES_ERR::
0224 330 CALCS #0,W^MAC$ERR_INTERN ;REPORT INTERNAL DIFFICULTIES
0224 331 BRW MAC$LAST_CHANCE ;STAY TUNED FOR QUICK EXIT
```

56 FDD9' 30 0000'CF 09 14 FDCA' 30 05 56 D7 50 0000'CF 56 D0 50 00000000'E6 9A 01 50 91 55 0000'CF 9E 14 11 55 0000'CF D0 36 13 OC A5 50 91 55 65 D0 07 F5 12 29 11 0000'CF 55 D0 0000'CF OC A5 9A 0000'CF OF A5 D0 50 00000000'EF 46 D0 05 13 0000'CF 50 D0 05 028D 327 028E 328 0000'CF 00 FB 028E 329 FD6A' 31 0293 331


```
0296 333 .SBTTL ALIGNMENT DIRECTIVE
0296 334
0296 335 :++
0296 336 : FUNCTIONAL DESCRIPTION:
0296 337 :
0296 338 : THIS DIRECTIVE CAUSES THE PC TO BE ALIGNED TO THE SPECIFIED
0296 339 : BOUNDARY, WITH THE OPTION OF THE SKIPPED BYTES BEING FILLED
0296 340 : WITH A FILL EXPRESSION.
0296 341 :
0296 342 :--
0296 343
0296 344 ALIGN::
00 6B 06 E5 0296 345 BBCC #FLGSV_EVAEXPR,(R11),+1 ;ALIGN HEAD = KALIGN
00 6B 07 E3 029A 346 BBCC #FLGSV_EXPOPT,(R11),+1 ;DON'T OUTPUT EXPRESSION
0000'CF D4 029E 347 CLRL W*MAC$GL_ABSFLAG ;ASSUME NICE EXPRESSION
FD5B' 30 02A2 348 BSBW MAC$SKIPSP ;ASSUME ABSOLUTE
04 E1 02A5 349 BBC #CHRSV_NUM BER,- ;SKIP SPACES
OE 0000'CA 02A7 350 W*MAC$AB_CMSK_TAB(R10),30$ ;BRANCH IF NOT NUMERIC
FD52' 30 02AB 351 BSBW MAC$DNUMBER ;YES--GET IT
50 0000'CF B0 02AE 352 MOVW W*MAC$GL_VALUE,R0 ;GET THE RESULT
51 01 50 78 02B3 353 ASHL R0,#1,R1 ;GET 2**(VALUE)
33 11 02B7 354 BRB 70$ ;FINISH UP
02B9 355
02B9 356 : SYMBOLIC ALIGNMENT
02B9 357
55 FD44' 30 02B9 358 30$: BSBW MAC$SYMSCNUP ;SCAN THE SYMBOL
18 50 E9 02BC 359 BLBC R0,50$ ;BRANCH IF NO SYMBOL SCANNED
00000000'EF 9E 02BF 360 MOVAB L*PSC$G_OPTIONS,R5 ;POINT TO PSECT OPTIONS
FD37' 30 02C6 361 BSBW MAC$SRC_LIST ;LOOK IT UP
OB 50 E9 02C9 362 BLBC R0,50$ ;BRANCH IF NOT FOUND
50 05 A1 D0 02CC 363 MOVL SYM$ VAL(R1),R0 ;GET THE BITS FOR SYMBOL
50 03FF 8F B3 02D0 364 BITW #PSC$M_ALLOPTNS,R0 ;Is this an alignment?
OC 13 02D5 365 BEQL 60$ ;IF EQL YES
02D7 366
02D7 367 : INVALID ALIGNMENT
02D7 368
02D7 369 50$: MAC_ERR INVALIDIGN ;Get message code
FD21' 30 02DC 370 BSBW MAC$ERRORLN ;ISSUE TO PASS 2
50 7C 02DF 371 CLRQ R0 ;DO NO ALIGNING
09 11 02E1 372 BRB 70$ ;...
02E3 373
02E3 374 : FINISH SYMBOL ALIGNMENT
02E3 375
50 50 0A EF 02E3 376 60$: EXTZV #PSC$V_ALIGNMENT,- ;GET EXPONENT VALUE FOR KEYWORD
51 01 50 78 02E5 377 #PSC$S_ALIGNMENT,R0,R0 ;CALCULATE ALIGNMENT FACTOR
52 0000'CF D0 02EC 378 ASHL R0,#1,R1 ;POINT TO CURRENT PSECT
0A EF 02F1 379 70$: MOVL W*MAC$GL_PSECTPTR,R2 ;GET PSECT ALIGNMENT
04 02F3 380 EXTZV #PSC$V_ALIGNMENT,-
04 02F4 381 #PSC$S_ALIGNMENT,-
53 OD A2 02F7 382 PSC$W_OPTIONS(R2),R3
54 01 53 78 02F7 383 ASHL R3,#1,R4 ;CALCULATE PSECT ALIGNMENT
54 51 D1 02FB 384 CMPL R1,R4 ;ALIGNMENT TOO BIG
OB 1B 02FE 385 BLEQU 80$ ;IF LEQU NO
0300 386 $MAC_ERR ALIGNXCEED ;Yes--set code
FCF8' 30 0305 387 BSBW MAC$ERRORLN ;ISSUE TO PASS 2
51 54 D0 0308 388 MOVL R4,R1 ;USE PSECT ALIGNMENT
0000'CF 51 D0 030B 389 80$: MOVL R1,W*MAC$GL_EXPOPV1 ;STORE VALUE FOR ALIGN1/ALIGN2
```

MAC\$APSECT
V04-000

PROCESS PSECT RELATED DIRECTIVES I 10
ALIGNMENT DIRECTIVE

16-SEP-1984 02:02:06 VAX/VMS Macro V04-00
5-SEP-1984 01:47:21 [MACRO.SRC]APSECT.MAR;1

Page 10
(7)

05 0310 390 RSB


```
0311 392 :++
0311 393 : FUNCTIONAL DESCRIPTION:
0311 394 :
0311 395 : ALIGN1/ALIGN2 ARE CALLED TO FINISH PROCESSING THE .ALIGN
0311 396 : DIRECTIVE. IF NECESSARY, THE PC IS ADJUSTED AND, IF THIS
0311 397 : IS ALIGN2, THEN THE FILL IS EMITTED TO PASS 2 ALSO.
0311 398 :
0311 399 :--
0311 400 :
0311 401 ALIGN1:: ;DIRECTIVE = ALIGN HEAD
06 6B 04 E5 0311 402 BBCC #FLG$V_DATRPT,(R11),ALIGN_COM ;FLAG NO FILL EXPRESSION
04 04 11 0315 403 BRB ALIGN_COM ;(YOU NEVER CAN TELL...)
0317 404 :
0317 405 ALIGN2:: ;DIRECTIVE = ALIGN_HEAD EXPR
0317 406 ;DIRECTIVE = ALIGN_HEAD DCOMMA EXPR
00 6B 04 E3 0317 407 BBCC #FLG$V_DATRPT,(R11),ALIGN_COM ;FLAG FILL EXPRESSION PRESENT
031B 408 ALIGN_COM:
0000'CF D5 031B 409 TSTL W^MAC$GL_ABSFLAG ;ABSOLUTE EXPRESSION?
08 13 031F 410 BEQL 10$ ;IF EQL YES
FCD7' 30 0321 411 $MAC_ERR INVALID ; No--get error code
52 0000'CF D0 0326 412 BSBW MAC$ERRORPT ;ISSUE ERROR TO PASS 2
46 13 032E 413 10$: MOVL W^MAC$GL_EXPOPV1,R2 ;GET THE ALIGNMENT FACTOR
53 52 D0 0330 414 BEQL 40$ ;IF EQL NONE
52 52 D7 0333 415 MOVL R2,R3 ;COPY IT
51 0000'CF D0 0335 416 DECL R2 ;
51 52 CA 0338 417 MCOML R2,R2 ;AND COMPLEMENT
51 52 CA 033D 418 MOVL W^MAC$GL_PC,R1 ;GET CURRENT PC
53 34 13 0340 419 BICL2 R2,R1 ;
53 51 C2 0342 420 BEQL 40$ ;IF EQL NO ADJUSTMENT NEEDED
0A 6B 04 E4 0345 421 SUBL2 R1,R3 ;
53 53 DD 0347 422 PUSHL R3 ;
19 11 034B 423 BBSC #FLG$V_DATRPT,(R11),20$ ;IS THERE A FILL?
0353 424 $INTOUT_LW INT$_AUGPC,R3 ;NO--AUGMENT PC
0355 425 BRB 30$ ;AND EXIT
0355 426 :
0355 427 : THERE IS A FILL
0355 428 :
0355 429 20$: $INTOUT_LW INT$_STKL,<W^MAC$AL_VALSTACK[R7]> ;STACK THE FILL EXPR
0360 430 $INTOUT_LW INT$_STKL,R3 ;STACK THE FILL COUNT
0368 431 $INTOUT_X INT$_STRB ;STORE REPEATED BYTE
53 8ED0 036E 432 30$: POPL R3 ;GET THE PC AUGMENTATION
0371 433 $INC_PC R3 ;ADJUST PC IN PASS 1 ALSO
05 0376 434 40$: RSB ;
0377 435 :
0377 436 .END
```

MAC\$APSECT
Symbol table

PROCESS PSECT RELATED DIRECTIVES

K 10

16-SEP-1984 02:02:06
5-SEP-1984 01:47:21

VAX/VMS Macro V04-00
[MACRO.SRC]APSECT.MAR;1

Page 12
(8)

```
$COUNT      = 0000003B
ALIGN        = 00000296 RG 03
ALIGN1       = 00000311 RG 03
ALIGN2       = 00000317 RG 03
ALIGN_COM    = 0000031B R 03
ARG$K_SIZE   = 000003E8
AUD$K_SIZE   = 00000010
BLNK         = 00000020
CHR$M_COMMA CR = 00000020
CHR$M_ILL CHR = 00000040
CHR$M_NUM BER = 00000010
CHR$M_SPA_MSK = 00000001
CHR$M_SYM_CH1 = 00000008
CHR$M_SYM_CHR = 00000004
CHR$M_SYM_DLM = 00000002
CHR$V_COMMA CR = 00000005
CHR$V_CVTLCW = 00000061
CHR$V_ILL CHR = 00000006
CHR$V_NOCVT   = 0000007F
CHR$V_NUM BER = 00000004
CHR$V_SPA_MSK = 00000000
CHR$V_SYM_CH1 = 00000003
CHR$V_SYM_CHR = 00000002
CHR$V_SYM_DLM = 00000001
CNT          = 00000001
CR           = 0000000D
CRESTO       = 00000224 RG 03
CSAVE        = 000001BD RG 03
ENB$G_LOCALSYMB ***** X 03
ERR          = 00000000
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$M_TOCFLG  = 00080000
FLG$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$V_OPTVFLIDX = 0000002C
FLG$V_ORDLST  = 00000011
FLG$V_P2      = 0000000E
FLG$V_RPTIRP  = 0000001C
FLG$V_SEQFIL  = 00000019
FLG$V_SKAN    = 0000000F
FLG$V_SPECOP  = 00000022
FLG$V_SPLALL  = 0000001A
FLG$V_STOIMF  = 00000012
FLG$V_SYM2COL = 0000002A
FLG$V_TOCFLG  = 00000013
FLG$V_UPAFLG  = 00000024
FLG$V_UPDFIL  = 00000027
FLG$V_UPMARG  = 00000026
FLG$V_XCRF    = 0000001F
HASH$Z        = 0000007F
HYPHEN        = 0000002D
INP$K_BUFSIZ  = 000003E8
INT$K_BUFSIZ  = 000013F4
INT$K_BUFWRN  = 00001390
INT$_ADD      = 00000001
INT$_AND      = 00000002
INT$_ASH      = 00000003
INT$_ASN      = 0000000C
INT$_AUGPC    = 0000000D
INT$_BDST     = 0000000E
INT$_CHKL     = 0000000F
INT$_DIV      = 00000004
INT$_END      = 00000010
INT$_EPT      = 00000011
INT$_ERR      = 00000012
INT$_ETX      = 00000013
INT$_FNEWL    = 00000014
INT$_ILG      = 00000000
INT$_INFO     = 0000003A
INT$_LGLAB    = 00000015
INT$_MACL     = 00000016
INT$_MUL      = 00000005
INT$_NEG      = 00000006
INT$_NEWL     = 00000017
INT$_NEWP     = 00000018
INT$_NOT      = 00000007
INT$_OP       = 00000019
INT$_OR       = 00000008
INT$_PRIL     = 0000001A
INT$_PRT      = 0000001B
INT$_PSECT    = 0000001C
INT$_REDEF    = 0000001D
INT$_REF      = 0000001E
INT$_REST     = 0000001F
INT$_SAME     = 00000009
INT$_SAVE     = 00000020
INT$_SBTTL    = 00000021
```


MAC\$APSECT
Symbol table

PROCESS PSECT RELATED DIRECTIVES

L 10

16-SEP-1984 02:02:06
5-SEP-1984 01:47:21VAX/VMS Macro V04-00
[MACRO.SRC]APSECT.MAR;1Page 13
(8)

INT\$ SETFLAG = 00000022
INT\$ SETLONG = 00000023
INT\$ SPIC = 00000024
INT\$ SPID = 00000025
INT\$ STIB = 00000026
INT\$ STIL = 00000028
INT\$ STIW = 00000027
INT\$ STKEPT = 00000029
INT\$ STKG = 0000002A
INT\$ STKL = 0000002B
INT\$ STKPC = 0000002C
INT\$ STKS = 0000002D
INT\$ STOB = 00000034
INT\$ STOL = 0000002E
INT\$ STOW = 00000035
INT\$ STRB = 0000002F
INT\$ STRL = 00000031
INT\$ STRSB = 00000032
INT\$ STRSW = 00000033
INT\$ STRW = 00000030
INT\$ STSB = 00000036
INT\$ STSW = 00000037
INT\$ SUB = 0000000A
INT\$ SUME = 00000039
INT\$ WRN = 00000038
INT\$ XOR = 0000000B
LST\$K_BUFSIZ = 00000086
LST\$K_L_P_PAGE = 0000003C
LST\$K_TITLE_SIZ = 00000028
MAC\$AB_CMSK_TAB ***** X 03
MAC\$AB_PSC_SBF ***** X 03
MAC\$AB_TMSYM ***** X 03
MAC\$ALC_2_PAGES ***** X 03
MAC\$AL_PSC_SLB ***** X 03
MAC\$AL_VALSTACK ***** X 03
MAC\$DNOMBER ***** X 03
MAC\$ERRORLN ***** X 03
MAC\$ERRORPT ***** X 03
MAC\$ERR_INTERN ***** X 03
MAC\$GETCHR ***** X 03
MAC\$GL_ABSFLAG ***** X 03
MAC\$GL_EXPOVL1 ***** X 03
MAC\$GL_LSB ***** X 03
MAC\$GL_PC ***** X 03
MAC\$GL_PSC_BLKPC ***** X 03
MAC\$GL_PSC_LIST ***** X 03
MAC\$GL_PSC_MAX ***** X 03
MAC\$GL_PSC_SBP ***** X 03
MAC\$GL_PSECT ***** X 03
MAC\$GL_PSECTPTR ***** X 03
MAC\$GL_VALUE ***** X 03
MAC\$G_SBNAM ***** X 03
MAC\$INTOUT_1_LW ***** X 03
MAC\$INTOUT_X ***** X 03
MAC\$LAST_CHANCE ***** X 03
MAC\$PSC_OPT_SCN 000000A3 R 03
MAC\$PSC_RES_ERR 0000028E RG 03

MAC\$SET_NEW_LSB ***** X 03
MAC\$SET_PC ***** X 03
MAC\$SKIPSP ***** X 03
MAC\$SKP_OPR ***** X 03
MAC\$SRC_LIST ***** X 03
MAC\$SYMSCNUP ***** X 03
MAC\$_ALIGNXCEED = 007D900A
MAC\$_DIRSYNX = 007D906A
MAC\$_INVALIGN = 007D912A
MAC\$_NOTPSECOPT = 007D919A
MAC\$_PSECBUFV = 007D91BA
MAC\$_PSECBUFUND = 007D91C2
MAC\$_PSECOPCNFL = 007D91B2
MAC\$_TOOMNYPSEC = 007D920A
MAC SUBSYS = 0000007D
OBJ\$K_BUFSIZ = 00000200
OPF\$M_LASTOPR = 00002000
OPF\$M_OPTEXP = 00001000
OPF\$V_LASTOPR = 0000000D
OPF\$V_OPTEXP = 0000000C
OPTION_CONFLICT 000001B5 R 03
OPTION_SCAN 000000BF R 03
PSC\$B_NAME 00000004
PSC\$B_SEG 0000000C
PSC\$B_UNUSED 0000000B
PSC\$G_OPTIONS ***** X 03
PSC\$K_BLKSI2 00000013
PSC\$K_NO_OPTNS = 0000000A
PSC\$L_CURLOC 0000000F
PSC\$L_LINK 00000000
PSC\$L_MAXLGTH 00000005
PSC\$M_ABS = FFFFFFFF7
PSC\$M_ALIGNFLG = 00004000
PSC\$M_ALLOPTNS = 000003FF
PSC\$M_BYTE = 00004000
PSC\$M_CON = FFFFFFFFB
PSC\$M_DEFAULT = 000001C8
PSC\$M_EXE = 000000C0
PSC\$M_GBL = 00000010
PSC\$M_LCL = FFFFFFFEF
PSC\$M_LIB = 00000002
PSC\$M_LONG = 00004800
PSC\$M_NOEXE = FFFFFFFBF
PSC\$M_NOPIC = FFFFFFFFE
PSC\$M_NORD = FFFFFFFF7F
PSC\$M_NOSHR = FFFFFFFDF
PSC\$M_NOVEC = FFFFFFFDFF
PSC\$M_NOWRT = FFFFFFFEFF
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 = FFFFFFFFD
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 00000000 RG 03
PSECT\$BLANK ***** X 03
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
SYM\$K_BLKSI2 0000000D
SYM\$K_MAXLEN = 0000001F
SYM\$K_TWOCOL = 00000010
SYM\$L_LINK 00000000
SYM\$L_VAL 00000005
SYM\$M_ABS = 00000010
SYM\$M_ASN = 00000100
SYM\$M_CRFO = 00002000
SYM\$M_DEBUG = 00000020
SYM\$M_DEF = 00000001
SYM\$M_DELMAC = 00000200
SYM\$M_EPT = 00000200
SYM\$M_EXTRN = 00000008
SYM\$M_GLOBL = 00000004
SYM\$M_LOCAL = 00000040
SYM\$M_ODBG = 00000400
SYM\$M_REF = 00000080
SYM\$M_RELPSECT = 00000800
SYM\$M_SUPR = 00004000
SYM\$M_WEAK = 00000002
SYM\$M_XCRF = 00001000
SYM\$V_ABS = 00000004
SYM\$V_ASN = 00000008
SYM\$V_CRFO = 0000000D

MAC\$APSECT
Symbol table

PROCESS PSECT RELATED DIRECTIVES

M 10

16-SEP-1984 02:02:06 VAX/VMS Macro V04-00
5-SEP-1984 01:47:21 [MACRO.SRC]APSECT.MAR;1

Page 14
(8)

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
SYMSV_FLAG = 00000009
TAB = 00000009
X1 = 00000400
X2 = 0000000F

+-----+
! 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_CODE_P1	00000377 (887.)	03 (3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.04	00:00:01.47
Command processing	128	00:00:00.37	00:00:04.68
Pass 1	208	00:00:03.42	00:00:14.44
Symbol table sort	0	00:00:00.41	00:00:01.64
Pass 2	95	00:00:00.91	00:00:04.17
Symbol table output	29	00:00:00.15	00:00:00.23
Psect synopsis output	2	00:00:00.01	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	499	00:00:05.32	00:00:26.65

The working set limit was 1350 pages.

32128 bytes (63 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 431 non-local and 45 local symbols.

436 source lines were read in Pass 1, producing 19 object records in Pass 2.

14 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	11
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	14

505 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

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

0224 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

