

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
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR		000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR		000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR		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
MMM		MMM	AAA	AAA	CCC	RRR	RRR	000
MMM		MMM	AAA	AAA	CCC	RRR	RRR	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

100

MAC

MAC

MAC
MAC

333

MAC
MAC

MAC

MAC
MAC

MAC

MAC
MACMAC
MACMAC
MAC

MAC

100

```

      AAAAAA      RRRRRRRR      GGGGGGGG      SSSSSSSS      CCCCCCCC      NN      NN
      AAAAAA      RRRRRRRR      GGGGGGGG      SSSSSSSS      CCCCCCCC      NN      NN
AA      AA      RR      RR      GG      SS      CC      NN      NN
AA      AA      RR      RR      GG      SS      CC      NN      NN
AA      AA      RR      RR      GG      SS      CC      NNNN      NN
AA      AA      RRRRRRRR      GG      SSSSSS      CC      NNNN      NN
AA      AA      RRRRRRRR      GG      SSSSSS      CC      NN      NN
AAAAA      RR      RR      GG      GG      SS      CC      NN      NN
AAAAA      RR      RR      GG      GG      SS      CC      NN      NN
AA      AA      RR      RR      GG      GG      SS      NN      NN
AA      AA      RR      RR      GG      GG      SS      NN      NN
AA      AA      RR      RR      GG      GG      SS      NN      NN
AA      AA      RR      RR      GG      GG      SS      NN      NN
      GG      SSSSSSSS      CCCCCCCC      NN      NN
      GG      SSSSSSSS      CCCCCCCC      NN      NN

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```


(2) 68
(3) 94

DECLARATIONS
MAC\$MAC_ARG_SCN SCAN MACRO KEYWORD/REAL ARGS

```
0000 1      .TITLE MAC$ARGSCN      SCAN MACRO REAL/KEYWORD ARGS
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 :      V02.08  CNH0047      Chris Hume      22-Dec-1980
0000 44 :      Count null argument after trailing comma for .NARG directive.
0000 45 :      (DEFINE.MAR 02.18, GETARG.MAR 02.06)
0000 46
0000 47 :      V01.07  RN0023      R. Newland      3-Nov-1979
0000 48 :      New error codes to get error message from system
0000 49 :      message file.
0000 50
0000 51 :      V01.06  RN0018      R. Newland      20-Oct-1979
0000 52 :      Convert argument to upper case if UPMARG set.
0000 53
0000 54 :      V01.05  RN0012      R. Newland      26-Sep-1979
0000 55 :      Fix problem with angle bracket processing of .IRPC
0000 56 :      string argument. SPR 11-25871
0000 57 :
```


0000	58 :	V01.04	RN0006	R. Newland	28-Aug-1979
0000	59 :				
0000	60 :				
0000	61 :				
0000	62 :	V01.03	RN0005	R. Newland	13-Aug-1979
0000	63 :				
0000	64 :				
0000	65 :				
0000	66 :--				

Fix problem with continuation lines on macro calls
SPR 11-24902

Remove .ALIGN LONG and DEBUG statements and use
symbolic value for maximum argument size

```
0000 68      .SBTTL  DECLARATIONS
0000 69      :
0000 70      : INCLUDE FILES:
0000 71      :
0000 72      :
0000 73      :
0000 74      : MACROS:
0000 75      :
0000 76      :
0000 77      $MAC_GENVALDEF      ;DEFINE GENERAL VALUES
0000 78      $MAC_MNBDEF        ;DEFINE MNB OFFSETS
0008 79      $MAC_CTLFLGDEF     ;DEFINE CONTROL FLAGS
0008 80      $MAC_SYMBLKDEF     ;DEFINE SYMBOL BLOCK OFFSETS
0000 81      $MACMSGDEF         ; Define message codes
0000 82      :
0000 83      : LOCAL DATA
0000 84      :
0000 85      :
00000000 86      .PSECT  MAC$RO_DATA,NOWRT,NOEXE,GBL, LONG
0000 87      :
0000 88      MAC$AB_SPLARGS:      ;SPECIAL MACRO ARG CHARACTERS
5E 3C 5C 0000 89      .BYTE   ^A/\/,^A/</,^A/^/
00000003 0003 90      MAC$K_SPLARG=-MAC$AB_SPLARGS
0003 91      :
00000000 92      .PSECT  MAC$RO_CODE_MAC,NOWRT,GBL, LONG
```



```
0000 94 .SBTTL MAC$MAC_ARG_SCN SCAN MACRO KEYWORD/REAL ARGS
0000 95
0000 96 :++
0000 97 : FUNCTIONAL DESCRIPTION:
0000 98
0000 99 : THIS ROUTINE SCANS A 'REAL' ARGUMENT AND ACCUMULATES IT IN
0000 100 : MAC$AB_TMPBUF. THE ROUTINE OPERATES AS FOLLOWS:
0000 101
0000 102 : 1) SKIP SPACES
0000 103 : 2) IF ARRIVE AT COMMA OR EOL RETURN NULL ARGUMENT
0000 104 : 3) IF ALPHABETIC CHARACTER AND KEYWORD ARGUMENTS PRESENT
0000 105 : (SET BY STAT4), SEE IF POSSIBLY KEYWORD ARG. IF IT
0000 106 : IS A KEYWORD ARG, STORE THE MAB ADDRESS IN MAC$GL_KEYPTR
0000 107 : AND SCAN PAST THE EQUAL SIGN.
0000 108 : 4) CHECK FOR SPECIAL CHARACTERS:
0000 109
0000 110 : A) "\" - THE VALUE OF THE SYMBOL FOLLOWING THE
0000 111 : BACKSLASH IS SUBSTITUTED. IF THE SYMBOL IS
0000 112 : UNDEFINED, A VALUE OF 0 IS SUBSTITUTED.
0000 113
0000 114 : B) "<" - THE STRING WITHIN THE MATCHED PAIR OF
0000 115 : ANGLE BRACKETS IS STORED. THE OUTER PAIR OF
0000 116 : ANGLE BRACKETS IS NOT STORED.
0000 117
0000 118 : C) "^" - THE CHARACTER FOLLOWING THE "^" IS TAKEN
0000 119 : AS THE DELIMITER. THE STRING WITHIN THE DELIMITERS
0000 120 : IS STORED.
0000 121
0000 122 : 5) IF THERE IS NO SPECIAL CHARACTER THE STRING IS COPIED
0000 123 : UNTIL A COMMA, EOL, OR RIGHT ANGLE BRACKET (IRP ONLY)
0000 124 : IS FOUND.
0000 125
0000 126 : OUTPUTS:
0000 127
0000 128 : R0 LENGTH OF ARGUMENT STORED IN MAC$AB_TMPBUF
0000 129
0000 130 :--
0000 131 :.ENABLE LSB
0000 132
0000 133 MAC$MAC_ARG_SCN::
0000 134 .PUSHR #^M<R6,R7,R8,R12> ;SAVE REGISTERS
0004 135
0000 136 CLRL W^MAC$GL_KEYPTR ;ASSUME NOT KEYWORD ARGUMENT
0004 137 MOVAB W^MAC$AB_TMPBUF,R8 ;POINT TO TEMP BUF
58 0000'CF D4 0004 138 MOVZWL #ARG$K_SIZE,R12 ;Set up temp buffer counter
5C 03E8 8F 3C 000D 139 BSBW MAC$SKIPSP ;SKIP SPACES
FFEB' 30 0012 140 CMPB R10,#CR ;GET TO EOL?
0D 5A 91 0015 141 BEQL 20$ ;IF EQL YES
12 13 0018 142 CMPB R10,#^A/,/ ;NO--STOP ON A COMMA?
2C 5A 91 001A 143 BNEQ 30$ ;IF NEQ NO
10 12 001D 144 BBS #FLG$V_IFSTAT,(R11),20$ ;YES--BRANCH IF IN AN IF
09 6B 17 E0 001F 145 BSBW MAC$GETCHR ;NO--SKIP THE COMMA
FFDA' 30 0023 146 BSBW MAC$SKIPSP ;AND SPACES
FFD7' 30 0026 147 BRW ARG_FOLLOWS ;Note trailing comma at exit.
0160 31 0029 148 20$: BRW ARG_SCAN_DONE ;GO FINISH UP
0163 31 002C 149
002F 150 : PICK UP FORMAL ARG NAME IF KEYWORD MACRO DEFINITION OR MACRO CALL.
```

```
002F 151 ; OTHERWISE TREAT AS POSITIONAL ARGUMENT.
002F 152 :
002F 153 30$: BBC #CHR$V_SYM CHR,- :CAN IT BE A SYMBOL?
4C 0000'CA E1 0031 154 W^MAC$AB_CMSK TAB(R10),70$ :BRANCH IF NO
55 0000'CF D0 0035 155 MOVL W^MAC$GL_KEYMAC,R5 ;YES--GET PTR TO KEYWORD ARGS
45 13 003A 156 BEQL 70$ ;IF EQL NO KEYWORD ARGS
5A DD 003C 157 PUSHL R10 ;SAVE INPUT CONTEXT
0000'CF DD 003E 158 PUSHL W^MAC$GL_LINEPT ;
7E 94 0042 159 CLRB -(SP) ;ASSUME NO CONTINUATION LINES
02 6B 03 E5 0044 160 BBCC #FLG$V_CONT,(R11),40$ ;CLEAR FLAG AND SEE IF IT WAS SET
6E 96 0048 161 INCB (SP) ;IT WAS--REMEMBER THAT
FFB3' 30 004A 162 40$: BSBW MAC$SYMSCNUP ;SCAN A SYMBOL
FFB0' 30 004D 163 BSBW MAC$SRC_LIST ;LOOK IT UP AMONG THE KEYWORDS
FFAD' 30 0050 164 BSBW MAC$SKIPSP ;Skip spaces after name
8E 95 0053 165 TSTB (SP)+ ;CHECK OLD CONT FLAG
04 13 0055 166 BEQL 50$ ;BRANCH IF CLEAR
00 6B 03 E3 0057 167 BBCC #FLG$V_CONT,(R11),.+1 ;ELSE SET IT AGAIN
55 50 7D 005B 168 50$: MOVQ R0,R5 ;COPY FLAG/ADDRESS
FF9F' 30 005E 169 BSBW MAC$SKIPSP ;SKIP SPACES
15 55 E9 0061 170 BLBC R5,60$ ;BRANCH IF KEYWORD NOT FOUND
3D 5A 91 0064 171 CMPB R10,#^A/=/ ;KEYWORD--FOLLOWED BY EQUAL SIGN?
10 12 0067 172 BNEQ 60$ ;IF NEQ NO
5E 08 C0 0069 173 ADDL2 #2*4,SP ;YES--THROW AWAY SAVED CONTEXT
0000'CF 56 D0 006C 174 MOVL R6,W^MAC$GL_KEYPTR ;REMEMBER KEYWORD ADDRESS
FF8C' 30 0071 175 BSBW MAC$GETCHR ;GET NEXT CHARACTER
FF89' 30 0074 176 BSBW MAC$SKIPSP ;SKIP SPACES
08 11 0077 177 BRB 70$
```



```
0079 179 :  
0079 180 : NOT A KEYWORD OR NO EQUAL SIGN FOLLOWING IT  
0079 181 :  
0000'CF 8ED0 0079 182 60$: POPL W^MAC$GL_LINEPT ;RESTORE SAVED CONTEXT  
5A 8ED0 007E 183 POPL R10 ;...  
0081 184 :  
0081 185 : LOOK FOR SPECIAL CHARACTERS THAT CAN START MACRO ARGS  
0081 186 :  
0000'CF 03 5A 3A 0081 187 70$: LOCC R10,#MAC$K_SPLARG,W^MAC$AB_SPLARGS ;SEE IF SPECIAL  
03 12 0087 188 BNEQ 90$ ;IF NEQ WE FOUND IT  
00A7 31 0089 189 80$: BRW 500$ ;NOT SPECIAL--GO PICK UP ARG  
50 51 00000000'8F C3 008C 190 90$: SUBL3 #MAC$AB_SPLARGS,R1,R0 ;COMPUTE INDEX  
02 00 50 8F 0094 191 CASEB R0,#0,#MAC$K_SPLARG-1 ;DISPATCH TO ROUTINE  
0009' 0098 192 100$: .WORD 200$-100$ ;"  
0035' 009A 193 .WORD 300$-100$ ;"  
0060' 009C 194 .WORD 400$-100$ ;"  
0092 31 009E 195 BRW 500$ ;THIS REALLY SHOULD NOT BE HAPPENING...  
00A1 196 ;BUT JUST TREAT AS NON-SPECIAL  
00A1 197 :  
00A1 198 : PROCESS SPECIAL CHARACTERS IN MACRO ARGUMENTS  
00A1 199 :  
00A1 200 :  
00A1 201 : CHARACTER IS A BACKSLASH. THE VALUE OF THE SYMBOL FOLLOWING THE  
00A1 202 : BACKSLASH IS SUBSTITUTED. IF THE SYMBOL IS NOT DEFINED A VALUE  
00A1 203 : OF 0 IS SUBSTITUTED.  
00A1 204 :  
FF5C' 30 00A1 205 200$: BSBW MAC$GETCHR ;SKIP THE BACKSLASH  
FF59' 30 00A4 206 BSBW MAC$SYMSCNUP ;SCAN THE SYMBOL  
20 50 E9 00A7 207 BLBC R0,230$ ;BRANCH IF NO SYMBOL FOUND  
FF53' 30 00AA 208 BSBW MAC$SRCUSRSYMTB ;LOOK IT UP  
05 50 E8 00AD 209 BLBS R0,210$ ;BRANCH IF FOUND  
51 50 FFFB'CF 9E 00B0 210 MOVAB W^MAC$GK_ZERO-SYMSL_VAL,R1 ;ELSE FAKE A ZERO  
50 05 A1 D0 00B5 211 210$: MOVL SYMSL_VAL(R1),R0 ;GET THE VALUE  
06 18 00B9 212 BGEQ 220$ ;BRANCH IF GEQ 0  
88 2D 90 00BB 213 MOVAB #^A/-/, (R8)+ ;NEGATIVE--EMIT A SIGN  
50 50 CE 00BE 214 MNEGL R0,R0 ;GET POSITIVE VALUE  
51 58 D0 00C1 215 220$: MOVL R8,R1 ;SET POINTER FOR DECOUT  
FF39' 30 00C4 216 BSBW MAC$DEC_OUT_L2X ;EMIT THE VALUE (DECIMAL)  
58 50 D0 00C7 217 MOVL R0,R8 ;GET UPDATED POINTER  
009D 31 00CA 218 230$: BRW 600$ ;FINISH UP
```

```
00CD 220 :  
00CD 221 : CHARACTER IS LEFT ANGLE BRACKET. THE STRING WITHIN THE ANGLE BRACKETS  
00CD 222 : IS COPIED INTO MAC$AB_TMPBUF.  
00CD 223 :  
6B 01 C8 00CD 224 300$: BISL2 #FLG$M_ALLCHR,(R11) :PASS SEMI COLONS  
55 D4 00D0 225 :CLRL R5 :NESTING LEVEL COUNTER STARTS AT 0  
FF2B' 30 00D2 226 310$: BSBW MAC$GETCHR :GET THE NEXT CHARACTER  
5C D7 00D5 227 :DECL R12 :ROOM TO STORE IT?  
03 19 00D7 228 :BLSS 315$ :IF LSS NO  
88 5A 90 00D9 229 :MOVB R10,(R8)+ :STORE IT  
3C 5A 91 00DC 230 315$: CMPB R10,#^A/</ :ANOTHER LEFT ANGLE BRACKET?  
04 12 00DF 231 :BNEQ 320$ :IF NEQ NO  
55 D6 00E1 232 :INCL R5 :YES--UP THE NESTING LEVEL  
ED 11 00E3 233 :BRB 310$ :KONTINUE SKANNING  
3E 5A 91 00E5 234 320$: CMPB R10,#^A/>/ :RIGHT ANGLE BRACKET?  
07 12 00E8 235 :BNEQ 330$ :IF NEQ NO  
E5 55 F4 00EA 236 :SOBGEQ R5,310$ :YES--DECREMENT NESTING LEVEL AND  
00ED 237 : :GO FOR NEXT CHAR IF NOT DONE  
58 D7 00ED 238 :DECL R8 :DONE--DON'T STORE FINAL '>'  
2B 11 00EF 239 :BRB 420$ :GO FINISH UP  
OD 5A 91 00F1 240 330$: CMPB R10,#CR :END OF LINE?  
DC 12 00F4 241 :BNEQ 310$ :IF NEQ NO--KEEP SKANNING  
24 11 00F6 242 :BRB 420$ :YES--FINISH UP  
00F8 243 :  
00F8 244 : CHARACTER IS UP-ARROW. THE CHARACTER FOLLOWING THE UP-ARROW IS TAKEN  
00F8 245 : AS THE DELIMITER. THE STRING WITHIN THE DELIMITERS IS COPIED INTO  
00F8 246 : MAC$AB_TMPBUF.  
00F8 247 :  
OD FF05' 30 00F8 248 400$: BSBW MAC$GETCHR :GET THE DELIMITER  
5A 91 00FB 249 :CMPB R10,#CR :IS IT EOL?  
24 13 00FE 250 :BEQL 440$ :IF EQL OOPS!  
55 5A D0 0100 251 :MOVL R10,R5 :SAVE THE DELIMITER  
6B 01 C8 0103 252 :BISL2 #FLG$M_ALLCHR,(R11) :AND PASS SEMI COLONS  
FEF7' 30 0106 253 410$: BSBW MAC$GETCHR :GET A CHARACTER  
55 5A 91 0109 254 :CMPB R10,R5 :IS IT THE DELIMITER?  
0E 13 010C 255 :BEQL 420$ :IF EQL YES  
OD 5A 91 010E 256 :CMPB R10,#CR :END OF LINE?  
09 13 0111 257 :BEQL 420$ :IF EQL YES (OOPS)  
5C D7 0113 258 :DECL R12 :NO--HAVE WE OVERFLOWED BUFFER?  
EF 19 0115 259 :BLSS 410$ :IF LSS YES--DO NOT STORE  
88 5A 90 0117 260 :MOVB R10,(R8)+ :NO--STORE THE CHAR  
EA 11 011A 261 :BRB 410$ :LOOP FOR MORE  
011C 262 :  
011C 263 : FINISH UP PROCESSING. RESET LOWER CASE LIMIT, AND CHECK FOR PROPER  
011C 264 : END OF LINE.  
011C 265 :  
6B 01 CA 011C 266 420$: BICL2 #FLG$M_ALLCHR,(R11) :DO NOT PASS SEMICOLONS  
OD 5A 91 011F 267 430$: CMPB R10,#CR :DID WE END WITH EOL?  
0A 12 0122 268 :BNEQ 450$ :IF NEQ NO--OK  
FED4' 30 0124 269 440$: $MAC_ERR UNTERMARG :Yes--that is an error  
3C 11 012C 270 :BSBW MAC$ERRORLN :ISSUE MESSAGE TO PASS 2  
FECF' 30 012E 271 :BRB 600$ :GO FINISH UP  
37 11 0131 272 450$: BSBW MAC$GETCHR :GET THE NEXT CHARACTER  
0133 273 :BRB 600$ :FINISH UP  
0133 274 :  
0133 275 : NOTHING SPECIAL--MUST BE SYMBOLIC ARGUMENT. COPY STRING UNTIL  
0133 276 : A COMMA, END OF LINE OR RIGHT ANGLE BRACKET (IRP ONLY) IS FOUND.
```



```
03 6B 5C D7 0133 277 :
      0A 19 0133 278 500$: DECL R12 ;ROOM TO STORE THE CHARACTER?
      26 E1 0135 279 BLSS 505$ ;IF LSS NO
      FEC2' 30 0137 280 BBC #FLGSV_UPMARG,(R11),502$ ; Branch if not to convert to upper case
      013B 281 BSBW MAC$CVT_LOWER ; Convert possible lower case to upper
      013E 282 502$:
      88 5A 90 013E 283 MOV B R10,(R8)+ ;COPY THE CHARACTER
      FEBC' 30 0141 284 505$: BSBW MAC$GETCHR ;GET A CHARACTER
      3E 5A 91 0144 285 CMP B R10,#^A/>/ ;RIGHT ANGLE BRACKET?
      08 12 0147 286 BNEQ 510$ ;IF NEQ NO
      04 6B 1C E1 0149 287 BBC #FLGSV_RPTIRP,(R11),510$ ; Branch if not .REPT, .IRP or .IRPC
      19 6B 1D E1 014D 288 BBC #FLGSV_IRPC,(R11),600$ ; Branch if not .IRPC
      2C 5A 91 0151 289 510$: CMP B R10,#^A/,/ ;IS IT A COMMA?
      14 13 0154 290 BEQL 600$ ;IF EQL YES--DONE
      0D 5A 91 0156 291 CMP B R10,#CR ;NO--END OF LINE?
      0F 13 0159 292 BEQL 600$ ;IF EQL YES
      29 5A 91 015B 293 CMP B R10,#^A/)/ ;RIGHT PARENS?
      04 12 015E 294 BNEQ 520$ ;IF NEQ NO
      06 6B 21 E0 0160 295 BBS #FLGSV_LEXOP,(R11),600$ ;YES--THAT IS TERMINATION IF LEXICAL OPERATO
      00 E1 0164 296 520$: BBC #CHRSV_SPA_MSK,- ;NO--KEEP GOING UNLESS A SPACE
      C9 0000'CA 0166 297 W^MAC$AB_CMSK_TAB(R10),500$
      016A 298 :
      016A 299 : ALL DONE (REALLY)
      016A 300 :
      5C D5 016A 301 600$: TSTL R12 ;WAS LINE TOO LONG?
      08 18 016C 302 BGEQ 605$ ;IF GEQ NO
      FEB8' 30 016E 303 $MAC_ERR_ARGTOOLONG ; Yes--get error code
      FEB7' 30 0173 304 BSBW MAC$ERRORLN ;ISSUE ERROR TO PASS 2
      2C 5A 91 0176 305 605$: BSBW MAC$SKIPSP ;SKIP SPACES
      14 12 017C 306 CMP B R10,#^A/,/ ;IS IT A COMMA?
      10 6B 17 E0 017E 307 BNEQ 610$ ;IF NEQ NO
      0C 6B 21 E0 0182 308 BBS #FLGSV_IFSTAT,(R11),610$ ;YES--BRANCH IF AN IF STATEMENT
      FE77' 30 0186 309 BBS #FLGSV_LEXOP,(R11),610$ ;OR IS THIS A LEXICAL OPERATOR?
      FE74' 30 0189 310 BSBW MAC$GETCHR ;NO--GET NEXT CHARACTER
      018C 311 BSBW MAC$SKIPSP ;SKIP SPACES
      018C 312 :
      06 6B 2D E2 018C 313 ARG_FOLLOWS: ; Note presence of trailing comma.
      04 11 0190 314 BBSS #FLGSV_MOREARG,(R11),615$
      0192 315 BRB 615$
      0192 316 610$:
      00 6B 2D E5 0192 317 ARG_SCAN DONE: ; Clear trailing comma flag.
      50 58 00000000'8F C3 0196 318 BBCC #FLGSV_MOREARG,(R11),615$
      11C0 8F BA 019E 319 615$: SUBL3 #MAC$AB_TMPBUF,R8,R0 ;FIGURE LENGTH OF ARG
      05 01A2 320 POPR #^M<R6,R7,R8,R12> ;SAVE REGISTERS
      01A3 321 RSB
      01A3 322
      01A3 323 .DISABLE LSB
      01A3 324 .END
```

MAC\$ARGSCN
Symbol table

SCAN MACRO REAL/KEYWORD ARGS

L 11

16-SEP-1984 02:02:40
5-SEP-1984 01:47:25VAX/VMS Macro V04-00
[MACRO.SRC]ARGSCN.MAR;1Page 9
(5)

ARG\$K_SIZE = 000003E8
ARG_FOLLOWS = 0000018C R
ARG_SCAN_DONE = 00000192 R
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
CHRS\$V_COMMA_CR = 00000005
CHRS\$V_CVT_LWC = 00000061
CHRS\$V_ILL_CHR = 00000006
CHRS\$V_NO_CVT = 0000007F
CHRS\$V_NUM_BER = 00000004
CHRS\$V_SPA_MSK = 00000000
CHRS\$V_SYM_CH1 = 00000003
CHRS\$V_SYM_CHR = 00000002
CHRS\$V_SYM_DLM = 00000001
CR = 0000000D
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

04
04

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 = 00008000
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
HASHSZ = 0000007F
HYPHEN = 0000002D
INP\$K_BUFSIZ = 000003E8
INT\$K_BUFSIZ = 000013F4
INT\$K_BUFWRN = 00001390
LST\$K_BUFSIZ = 00000086
LST\$K_L_P_PAGE = 0000003C
LST\$K_TITLE_SIZE = 00000028
MAB\$B_ARGNO = 00000005
MAB\$B_NAME = 00000004
MAB\$K_BLKSIZ = 0000000C
MAB\$S_DVPTR = 00000008
MAB\$S_LINK = 00000000
MAB\$S_DVLEN = 00000006
MAC\$AB_CMSK_TAB ***** X 04
MAC\$AB_SPLARGS 00000000 R 03
MAC\$AB_TMPBUF ***** X 04
MAC\$CVT_LOWER ***** X 04
MAC\$DEC_OUT_L2X ***** X 04
MAC\$ERRORLN ***** X 04
MAC\$GETCHR ***** X 04
MAC\$GK_ZERO ***** X 04
MAC\$GL_KEYMAC ***** X 04
MAC\$GL_KEYPTR ***** X 04
MAC\$GL_LINEPT ***** X 04
MAC\$K_SPLARG = 00000003
MAC\$MAC_ARG_SCN 00000000 RG 04
MAC\$SKIPSP ***** X 04
MAC\$SRCUSRSYMTB ***** X 04
MAC\$SRC_LIST ***** X 04
MAC\$SYM\$CNUP ***** X 04
MAC\$_ARGTOOLONG = 007D9012
MAC\$_UNTERMARG = 007D922A
MAC SUBSYS = 0000007D
MNB\$B_ARGCT 00000017
MNB\$B_NAME 00000004
MNB\$K_BLKSIZ 0000001C
MNB\$S_ARGP 00000018
MNB\$S_CRSYMF 00000013
MNB\$S_LINK 00000000
MNB\$S_PAGC 0000000F
MNB\$S_PAGP 0000000B
MNB\$S_TXTP 00000005
MNB\$S_FLAG 00000009
MXB\$K_BLKSIZ 00000008
MXB\$S_LINK 00000000

MAC\$ARGSCN
Symbol table

SCAN MACRO REAL/KEYWORD ARGS

M 11

16-SEP-1984 02:02:40
5-SEP-1984 01:47:25

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

Page 10
(5)

MXBSL_PAGES = 00000004
 OBJ\$K_BUF\$IZ = 00000200
 OPFSM_LASTOPR = 00002000
 OPFSM_OPTEXP = 00001000
 OPFSV_LASTOPR = 0000000D
 OPFSV_OPTEXP = 0000000C
 PSC\$B_NAME = 00000004
 PSC\$B_SEG = 0000000C
 PSC\$B_UNUSED = 0000000B
 PSC\$K_BLK\$IZ = 00000013
 PSC\$K_NO_OPTNS = 0000000A
 PSC\$K_CURLOC = 0000000F
 PSC\$K_LINK = 00000000
 PSC\$K_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 = FFFFFFF7F
 PSC\$M_NOSHR = FFFFFFFDF
 PSC\$M_NOVEC = FFFFFFFDF
 PSC\$M_NOWRT = FFFFFFFEF
 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

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_BLK\$IZ = 0000000D
 SYM\$K_MAXLEN = 0000001F
 SYM\$K_TWOCOL = 00000010
 SYM\$K_LINK = 00000000
 SYM\$K_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
 SYM\$V_DEBUG = 00000005
 SYM\$V_DEF = 00000000
 SYM\$V_DELMAC = 00000009
 SYM\$V_EPT = 00000009
 SYM\$V_EXTRN = 00000003
 SYM\$V_GLOBL = 00000002
 SYM\$V_LOCAL = 00000006
 SYM\$V_ODBG = 0000000A
 SYM\$V_REF = 00000007
 SYM\$V_RELPSECT = 0000000B
 SYM\$V_SUPR = 0000000E
 SYM\$V_WEAK = 00000001
 SYM\$V_XCRF = 0000000C
 SYM\$W_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
\$AB\$\$	0000001C (28.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC\$RO_DATA	00000003 (3.)	03 (3.)	NOPIC USR CON REL GBL NOSHR NOEXE RD NOWRT NOVEC LONG
MAC\$RO_CODE_MAC	000001A3 (419.)	04 (4.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:01.31
Command processing	103	00:00:00.35	00:00:01.89
Pass 1	186	00:00:02.57	00:00:15.81
Symbol table sort	0	00:00:00.35	00:00:00.60
Pass 2	79	00:00:00.67	00:00:03.66
Symbol table output	23	00:00:00.12	00:00:00.69
Psect synopsis output	3	00:00:00.03	00:00:00.18
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	425	00:00:04.12	00:00:24.14

The working set limit was 1350 pages.

22809 bytes (45 pages) of virtual memory were used to buffer the intermediate code.

There were 20 pages of symbol table space allocated to hold 361 non-local and 33 local symbols.

324 source lines were read in Pass 1, producing 17 object records in Pass 2.

10 pages of virtual memory were used to define 9 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	6
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	9

418 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

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

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

