```
MMM          MMM     AAAAAAAAA         CCCCCCCCCCC   RRRRRRRRRRR        OOOOOOOOO
MMM          MMM     AAAAAAAAA         CCCCCCCCCCC   RRRRRRRRRRR        OOOOOOOOO
MMM          MMM     AAAAAAAAA         CCCCCCCCCCC   RRRRRRRRRRR        OOOOOOOOO
MMMMMM    MMMMMM   AAA         AAA   CCC            RRR         RRR   OOO         OOO
MMMMMM    MMMMMM   AAA         AAA   CCC            RRR         RRR   OOO         OOO
MMMMMM    MMMMMM   AAA         AAA   CCC            RRR         RRR   OOO         OOO
MMM   MMM   MMM    AAA         AAA   CCC            RRR         RRR   OOO         OOO
MMM   MMM   MMM    AAA         AAA   CCC            RRR         RRR   OOO         OOO
MMM   MMM   MMM    AAA         AAA   CCC            RRR         RRR   OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRRRRRRRRRR       OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRRRRRRRRRR       OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRRRRRRRRRR       OOO         OOO
MMM         MMM    AAAAAAAAAAAAAAA   CCC            RRR   RRR         OOO         OOO
MMM         MMM    AAAAAAAAAAAAAAA   CCC            RRR   RRR         OOO         OOO
MMM         MMM    AAAAAAAAAAAAAAA   CCC            RRR   RRR         OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRR      RRR      OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRR      RRR      OOO         OOO
MMM         MMM    AAA         AAA   CCC            RRR      RRR      OOO         OOO
MMM         MMM    AAA         AAA      CCCCCCCCCCC   RRR         RRR      OOOOOOOOO
MMM         MMM    AAA         AAA      CCCCCCCCCCC   RRR         RRR      OOOOOOOOO
MMM         MMM    AAA         AAA      CCCCCCCCCCC   RRR         RRR      OOOOOOOOO
```

FILEID**CRFSUB

```
  CCCCCCCC  RRRRRRRR    FFFFFFFFFF    SSSSSSSS  UU      UU  BBBBBBBB
  CCCCCCCC  RRRRRRRR    FFFFFFFFFF    SSSSSSSS  UU      UU  BBBBBBBB
CC          RR      RR  FF            SS        UU      UU  BB      BB
CC          RR      RR  FF            SS        UU      UU  BB      BB
CC          RR      RR  FF            SS        UU      UU  BB      BB
CC          RR      RR  FF            SS        UU      UU  BB      BB
CC          RRRRRRRR    FFFFFFFF        SSSSSS  UU      UU  BBBBBBBB
CC          RRRRRRRR    FFFFFFFF        SSSSSS  UU      UU  BBBBBBBB
CC          RR  RR      FF                    SS  UU      UU  BB      BB
CC          RR  RR      FF                    SS  UU      UU  BB      BB
CC          RR    RR    FF                    SS  UU      UU  BB      BB
CC          RR    RR    FF                    SS  UU      UU  BB      BB
  CCCCCCCC  RR      RR  FF            SSSSSSSS    UUUUUUUUUU  BBBBBBBB
  CCCCCCCC  RR      RR  FF            SSSSSSSS    UUUUUUUUUU  BBBBBBBB

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

Table of contents

```
0000     1          .TITLE  MAC$CRFSUB      SUBROUTINES FOR USE WITH CREF/MACRO
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: 28-AUG-78
0000    40  ;
0000    41  ; MODIFIED BY:
0000    42  ;
0000    43  ;       V01.06  RN0024          R. Newland              13-Dec-1979
0000    44  ;               Change cref output routine argument processing.
0000    45  ;
0000    46  ;       V01.05  RN0005          R. Newland      13-Aug-1979
0000    47  ;               Variable symbol name storage and remove .ALIGN LONG statments
0000    48  ;
0000    49  ;       V01.04  RN0002          R. Newland      01-Feb-1979
0000    50  ;               Changes for Source Update Merge, convert line number
0000    51  ;               into update format for updated line
0000    52  ;--
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; INCLUDE FILES:
                ;

                ;
                ; MACROS:
                ;
                        $MAC_CTLFLGDEF                  ;DEFINE CONTROL FLAGS
                        $MAC_SYMBLKDEF                  ;DEFINE SYMBOL BLOCK OFFSETS
                        $MAC_CRFLAGDEF                  ;DEFINE CREF CONTROL FLAGS
                        $MAC_MNBDEF                     ;DEFINE MACRO NAME BLOCK
                        $MAC_OPRDEF                     ;DEFINE OPERAND DESCRIPTOR BITS
                        $RABDEF                         ;DEFINE RAB OFFSETS
                        DEFSUMCBL                       ; Define SUM control block symbols

00000000                .PSECT  MAC$RO_CODE_P15,NOWRT,GBL,LONG
```

```
                                0000     73          .SBTTL  MAC$CVT_LIN_PAG CONVERT LINE/PAGE TO ASCII FOR CREF
                                0000     74
                                0000     75 ;++
                                0000     76 ; FUNCTIONAL DESCRIPTION:
                                0000     77 ;
                                0000     78 ;       THIS ROUTINE CONVERTS THE CURRENT LINE/PAGE NUMBER INTO ASCII
                                0000     79 ;       IN MAC$AB_LPBUF TO USE WHEN CALLING CREF WITH A REFERENCE
                                0000     80 ;       OR DEFINITION.
                                0000     81 ;
                                0000     82 ;       THE LINE AND PAGE NUMBERS ARE RETURNED IN THE FOLLOWING FORMAT:
                                0000     83 ;
                                0000     84 ;               LLLLL (PPPP)
                                0000     85 ;
                                0000     86 ;--
                                0000     87
                                0000     88 MAC$CVT_LIN_PAG::
                0000'CF   9F    0000     89          PUSHAB  W^MAC$AB_LPBUF          ;STACK BUFFER ADDRESS
00 BE   10   20 6E     00   2C 0004     90          MOVC5   #0,(SP),#^A/ /,#16,@(SP) ;BLANK FILL THE BUFFER
             51 6E     D0   000B     91          MOVL    (SP),R1                 ;POINT TO THE BUFFER
                81     94   000E     92          CLRB    (R1)+                   ;LEAVE ROOM FOR COUNT BYTE
             50 0000'CF   3C   0010     93          MOVZWL  W^MAC$GW_LST_LINE,R0    ; Get listing line number
               14 6B    27   E1   0015     94          BBC     #FLG$V_UPDFIL,(R11),10$ ; Branch if input file is not being updated
            001C'CF     02   E1   0019     95          BBC     #SUM_V_SRCUPD, -        ; Branch if line is not an update
                       0E    001E     96                  W^MAC$GT_SCB+SUM_B_FLAGS,10$
                  FFDE'  30   001F     97          BSBW    MAC$DEC_OUT_L2X         ; Convert line number
                51   50   D0   0022     98          MOVL    R0,R1                   ; Reset character output pointer
                81   2E   90   0025     99          MOVB    #^A/./,(R1)+            ; Insert separating '.'
             50 0000'CF   3C   0028    100          MOVZWL  W^MAC$GW_LST_INST,R0    ; Get insert number
                  FFD0'  30   002D    101 10$:     BSBW    MAC$DEC_OUT_L2X         ;CONVERT LINE NUMBER
             51 6E    07   C1   0030    102          ADDL3   #7,(SP),R1              ;SKIP TO PAGE # FIELD
             81 2820 8F   B0   0034    103          MOVW    #^A/ (/,(R1)+           ;START THE PAGE NUMBER
             50 0000'CF   D0   0039    104          MOVL    W^MAC$GL_SRCPAG,R0      ;GET SOURCE PAGE NUMBER
                  FFBF'  30   003E    105          BSBW    MAC$DEC_OUT_L2X         ;CONVERT PAGE NUMBER
                80   29   90   0041    106          MOVB    #^A/)/,(R0)+            ;FINISH PAGE NUMBER
                   51 8ED0   0044    107          POPL    R1                      ;POINT TO BUFFER START AGAIN
                50   51   C2   0047    108          SUBL2   R1,R0                   ;FIGURE LENGTH OF STRING
             61 50   01   83   004A    109          SUBB3   #1,R0,(R1)              ;GET REAL LENGTH AND STORE IT
                          05   004E    110          RSB
```

```
                                004F   112          .SBTTL  MAC$CREF_ERROR  HANDLER FOR CREF ERRORS
                                004F   113
                                004F   114 ;++
                                004F   115 ; FUNCTIONAL DESCRIPTION:
                                004F   116 ;
                                004F   117 ;       THIS ROUTINE IS CALLED FROM CREF ON AN ERROR
                                004F   118 ;
                                004F   119 ;--
                                004F   120
                                004F   121 MAC$CRF_ERROR::
                           0000 004F   122          .WORD   0                       ;REGISTER SAVE MASK
                0000'CF    50  D0 0051   123          MOVL    R0,W^MAC$GL_STATUS      ;SET LATEST STATUS FOR EXIT
                          00   DD 0056   124          PUSHL   #0                      ;NO FAO ARGUMENTS
                          50   DD 0058   125          PUSHL   R0                      ;STACK THE ERROR CODE
                          02   DD 005A   126          PUSHL   #2                      ;THERE ARE TWO LONGWORDS IN PACKET
                     0000'CF   7F 005C   127          PUSHAQ  W^MAC$MACRO_NAME        ;STACK ADDRESS OF FACILITY NAME
                          7E   D4 0060   128          CLRL    -(SP)                   ;ZERO ADDRESS OF ACTION ROUTINE
                       08 AE   9F 0062   129          PUSHAB  8(SP)                   ;PUSH ADDRESS OF MESSAGE BUFFER
           00000000'GF    03   FB 0065   130          CALLS   #3,G^SYS$PUTMSG         ;PRINT THE CREF ERROR
                     FF91'    31 006C   131          BRW     MAC$LAST_CHANCE         ;GO DIE UNGRACEFULLY
                                006F   132
                                006F   133          .SBTTL  MAC$CRF_OUTPUT  OUTPUT CROSS-REFERENCE LINE
                                006F   134
                                006F   135 ;++
                                006F   136 ; FUNCTIONAL DESCRIPTION:
                                006F   137 ;
                                006F   138 ;       THIS ROUTINE IS CALLED BY CREF TO OUTPUT A LINE:
                                006F   139 ;
                                006F   140 ; CALLING SEQUENCE:
                                006F   141 ;
                                006F   142 ;       CALLS   #1,MAC$CRF_OUTPUT
                                006F   143 ;
                                006F   144 ; INPUTS:
                                006F   145 ;
                                006F   146 ;       4(AP)   Address of line descriptor
                                006F   147 ;
                                006F   148 ;--
                                006F   149
                                006F   150 MAC$CRF_OUTPUT::
                           0800 006F   151          .WORD   ^M<R11>                 ;IN CASE WE NEED TO DO NEW PAGE
          31 0000'CF    09   E1 0071   152          BBC     #FLG$V_LSTXST,W^MAC$GL_FLAGS,20$ ;BRANCH IF NO LISTING FILE!
                0000'CF    D5 0077   153          TSTL    W^MAC$GL_LINE_CNT       ;TIME FOR NEW PAGE?
                          08   14 007B   154          BGTR    10$                     ;IF GTR NO
             5B   0000'CF    9E 007D   155          MOVAB   W^MAC$GL_FLAGS,R11      ;ENSURE R11 POINTS TO FLAGS WORD
                     FF7B'    30 0082   156          BSBW    MAC$LST_PAG_HDR         ;YES--OUTPUT PAGE HEADER
                0000'CF    D7 0085   157 10$:       DECL    W^MAC$GL_LINE_CNT       ;DEC. LINES LEFT ON PAGE
             50   0000'CF    9E 0089   158          MOVAB   W^MAC$LIST_RAB,R0       ;POINT TO LISTING RAB
                51    04 AC   D0 008E   159          MOVL    4(AP),R1                ; Get address of line descriptor
                22 A0   61   B0 0092   160          MOVW    (R1),RAB$W_RSZ(R0)      ; Set record size
             28 A0    04 A1   D0 0096   161          MOVL    4(R1),RAB$L_RBF(R0)     ; and record address
                                009B   162          $PUT    RAB=(R0),-              ;WRITE LINE TO LISTING
                                009B   163                  ERR=W^MAC$ERR_PUT
                          04   00A8   164 20$:       RET
```

```
                                00A9   166          .SBTTL  MAC$CREF_SYM    EMIT CROSS REFERENCE INFO FOR SYMBOL
                                00A9   167
                                00A9   168 ;++
                                00A9   169 ; FUNCTIONAL DESCRIPTION:
                                00A9   170 ;
                                00A9   171 ;       THIS ROUTINE IS CALLED TO OUTPUT CROSS REFERENCE INFORMATION
                                00A9   172 ;       FOR A SYMBOL TO THE CROSS-REFERENCER.
                                00A9   173 ;
                                00A9   174 ; INPUTS:
                                00A9   175 ;
                                00A9   176 ;       R5      CRF$K_REF OR CRF$K_DEF
                                00A9   177 ;       R6      SYMBOL BLOCK ADDRESS
                                00A9   178 ;
                                00A9   179 ;--
                                00A9   180
                                00A9   181 MAC$CREF_SYM::
                    04     E1   00A9   182          BBC     #CRF$V_SYMBOLS,-        ;BRANCH IF NOT CREFFING SYMBOLS
                63 0000'CF         00AB   183                  W^MAC$GL_CRF_FLG,30$    ;...
                5F 6B    1F    E0  00AF   184          BBS     #FLG$V_XCRF,(R11),30$   ;BRANCH IF .NOCROSS IN EFFECT
             5A 09 A6    06    E0  00B3   185          BBS     #SYM$V_LOCAL,SYM$W_FLAG(R6),30$ ;DON'T CREF LOCAL SYMBOLS
             55 09 A6    0C    E0  00B8   186          BBS     #SYM$V_XCRF,SYM$W_FLAG(R6),30$ ;BRANCH IF .NOCROSS THIS SYMBOL
                                00BD   187 ;*****
                                00BD   188 ; NEXT INSTRUCTION DEPENDS ON CRF$K_DEF=1
                                00BD   189 ;*****
                       05 55    E8  00BD   190          BLBS    R5,5$                   ;BRANCH IF DEFINITION
             1A 09 A6    0D    E2  00C0   191          BBSS    #SYM$V_CRFO,SYM$W_FLAG(R6),10$ ;BRANCH IF INSERT KEY DONE
                7E    09 A6    3C  00C5   192 5$:     MOVZWL  SYM$W_FLAG(R6),-(SP)    ;NO--INSERT KEY NOW
                      05 A6    9F  00C9   193          PUSHAB  SYM$L_VAL(R6)           ;STACK ADDRESS OF VALUE
                   50    04 A6    9A  00CC   194          MOVZBL  SYM$B_NAME(R6),R0       ; Get offset to counted name
                 7E    56    50    C3  00D0   195          SUBL3   R0,R6,-(SP)             ; and form its address on stack
                      0000'CF    9F  00D4   196          PUSHAB  W^MAC$AL_CRFSYCTB       ;STACK CONTROL TABLE ADDRESS
          00000000'GF       04    FB  00D8   197          CALLS   #4,G^CRF$INSRTKEY       ;INSERT THE KEY IN CREF TABLE
                            55    DD  00DF   198 10$:    PUSHL   R5                      ;STACK REF/DEF FLAG
                            00    DD  00E1   199          PUSHL   #0                      ;ASSUME A READ REFERENCE
                   50   0000'CF    D0  00E3   200          MOVL    W^MAC$GL_MOPPTR,R0      ;GET POINTER TO OPERAND BYTES
                            0A    13  00E8   201          BEQL    20$                     ;IF EQL NOT INSTRUCTION--MUST BE READ
                   60   0060 8F    B3  00EA   202          BITW    #OPD$M_MODIFY!OPD$M_WRITE,(R0) ;IS IT MODIFY OR WRITE?
                            03    13  00EF   203          BEQL    20$                     ;IF EQL NO--NOT DESTRUCTIVE
                         6E    01    D0  00F1   204          MOVL    #1,(SP)                 ;YES--MARK FLAGS AS SUCH
                         0000'CF    9F  00F4   205 20$:     PUSHAB  W^MAC$AB_LPBUF          ;STACK REFERENCER NAME ADDRESS
                      50    04 A6    9A  00F8   206          MOVZBL  SYM$B_NAME(R6),R0       ; Get offset to counted name
                   7E    56    50    C3  00FC   207          SUBL3   R0,R6,-(SP)             ; and form its address on stack
                         0000'CF    9F  0100   208          PUSHAB  W^MAC$AL_CRFSYCTB       ;STACK CONTROL TABLE ADDRESS
                            FEF9    30  0104   209          BSBW    MAC$CVT_LIN_PAG         ;CONVERT LINE/PAGE INTO LPBUF
          00000000'GF       05    FB  0107   210          CALLS   #5,G^CRF$INSRTREF       ;INSERT REF/DEF INTO CREF TABLE
                         0000'CF    D6  010E   211          INCL    W^MAC$GL_CRF_CNT        ;COUNT SYMBOL AS CREFFED
                                05    0112   212 30$:     RSB
```

```
                     0113   214          .SBTTL  MAC$CREF_MACRO  EMIT CROSS REFERENCE INFO FOR MACRO
                     0113   215
                     0113   216 ;++
                     0113   217 ; FUNCTIONAL DESCRIPTION:
                     0113   218 ;
                     0113   219 ;       THIS ROUTINE IS CALLED TO OUTPUT CROSS REFERENCE INFORMATION FOR
                     0113   220 ;       A MACRO TO THE CROSS-REFERENCER.
                     0113   221 ;
                     0113   222 ; INPUTS:
                     0113   223 ;
                     0113   224 ;       R5      CRF$K_REF OR CRF$K_DEF
                     0113   225 ;       R6      MACRO NAME BLOCK ADDRESS
                     0113   226 ;
                     0113   227 ;--
                     0113   228
                     0113   229 MAC$CREF_MACRO::
          01   E1    0113   230          BBC     #CRF$V_MACROS,-         ;BRANCH IF NOT CREFFING MACROS
                               W^MAC$GL_CRF_FLG,30$    ;...
      45 0000'CF       1F   0115   231
      41 6B    1F   E0  0119   232          BBS     #FLG$V_XCRF,(R11),30$   ;BRANCH IF .NOCROSS IN EFFECT
   1A 09 A6    0D   E2  011D   233          BBSS    #SYM$V_CRFO,MNB$W_FLAG(R6),10$ ;BRANCH IF INSERT DONE THIS MACRO
     7E    09 A6    3C  0122   234          MOVZWL  MNB$W_FLAG(R6),-(SP)    ;NO--INSERT KEY NOW
        0F A6    9F  0126   235          PUSHAB  MNB$L_PAGC(R6)          ;STACK ADDRESS OF PAGE COUNT
      50    04 A6    9A  0129   236          MOVZBL  MNB$B_NAME(R6),R0       ; Get offset to counted name
     7E    56    50   C3  012D   237          SUBL3   R0,R6,-(SP)             ; and form its address on stack
        0000'CF    9F  0131   238          PUSHAB  W^MAC$AL_CRFMCCTB       ;STACK CONTROL TABLE ADDRESS
00000000'GF    04   FB  0135   239          CALLS   #4,G^CRF$INSRTKEY       ;INSERT KEY INCREF TABLE
           55   DD  013C   240 10$:    PUSHL   R5                      ;STAKC REF/DEF FLAG
           00   DD  013E   241          PUSHL   #0                      ;PUSH REFERENCER FLAGS (NYI)
        0000'CF    9F  0140   242          PUSHAB  W^MAC$AB_LPBUF          ;STACK REFERENCER NAME ADDRESS
      50    04 A6    9A  0144   243          MOVZBL  MNB$B_NAME(R6),R0       ; Get offset to counted name
     7E    56    50   C3  0148   244          SUBL3   R0,R6,-(SP)             ; and form its address on stack
        0000'CF    9F  014C   245          PUSHAB  W^MAC$AL_CRFMCCTB       ;STACK CONTROL TABLE ADDRESS
          FEAD   30  0150   246          BSBW    MAC$CVT_LIN_PAG         ;CONVERT LINE/PAGE INTO LPBUF
00000000'GF    05   FB  0153   247          CALLS   #5,G^CRF$INSRTREF       ;INSERT REF/DEF INTO CREF TABLE
        0000'CF    D6  015A   248          INCL    W^MAC$GL_CRF_MCNT       ;COUNT MACRO AS CREFFED
           05   05  015E   249 30$:    RSB
```

```
                                015F    251          .SBTTL  MAC$CREF_OPCODE EMIT CREF INFO FOR OPCODE
                                015F    252
                                015F    253 ;++
                                015F    254 ; FUNCTIONAL DESCRIPTION:
                                015F    255 ;
                                015F    256 ;       THIS ROUTINE IS CALLED TO OUTPUT CREF INFORMATION FOR AN
                                015F    257 ;       OPCODE.
                                015F    258 ;
                                015F    259 ; INPUTS:
                                015F    260 ;
                                015F    261 ;       R6      POINTS TO SYMBOL BLOCK
                                015F    262 ;
                                015F    263 ;--
                                015F    264
                                015F    265 MAC$CREF_OPCODE::
     42 0000'CF    02    E1     015F    266          BBC     #CRF$V_OPCODES,W^MAC$GL_CRF_FLG,30$ ;BRANCH IF NOT CREFFING OPCODES
        3E 6B     1F    E0     0165    267          BBS     #FLG$V_XCRF,(R11),30$   ;BRANCH IF .NOCROSS IN EFFECT
                                0169    268 ;
                                0169    269 ; BECAUSE THE OPCODE TABLE IS IN WRITE-ONLY MEMORY, WE MUST ALWAYS DO
                                0169    270 ; AN INSERT KEY. SORRY ABOUT THAT, FOLKS.
                                0169    271 ;
                  00    DD      0169    272          PUSHL   #0                      ;STACK FLAGS
                05 A6    9F     016B    273          PUSHAB  SYM$L_VAL(R6)           ;STACK VALUE
           50   04 A6    9A     016E    274          MOVZBL  SYM$B_NAME(R6),R0       ; Get offset to counted name
        7E   56   50    C3      0172    275          SUBL3   R0,R6,-(SP)             ; and form its address on stack
             0000'CF    9F     0176    276          PUSHAB  W^MAC$AL_CRFOPCTB       ;STACK CONTROL TABLE ADDRESS
    00000000'GF    04    FB     017A    277          CALLS   #4,G^CRF$INSRTKEY       ;INSERT THE KEY
         00000000'8F    DD     0181    278          PUSHL   #CRF$K_REF              ;THIS IS A REFERENCE
                  00    DD      0187    279          PUSHL   #0                      ;REFERENCER FLAGS
             0000'CF    9F     0189    280          PUSHAB  W^MAC$AB_LPBUF          ;STACK REFERENCER NAME ADDRESS
           50   04 A6    9A     018D    281          MOVZBL  SYM$B_NAME(R6),R0       ; Get offset to counted name
        7E   56   50    C3      0191    282          SUBL3   R0,R6,-(SP)             ; and form its address on stack
             0000'CF    9F     0195    283          PUSHAB  W^MAC$AL_CRFOPCTB       ;STACK CONTROL TABLE ADDRESS
                FE64    30     0199    284          BSBW    MAC$CVT_LIN_PAG         ;CONVERT LINE/PAGE INTO LPBUF
    00000000'GF    05    FB     019C    285          CALLS   #5,G^CRF$INSRTREF       ;INSERT REFERENCE
             0000'CF    D6     01A3    286          INCL    W^MAC$GL_CRF_OCNT       ;COUNT OPCODE REFERENCE
                       05     01A7    287 30$:     RSB
```

```
                                01A8   289          .SBTTL  MAC$CREF_DIR    EMIT CREF INFO FOR DIRECTIVE
                                01A8   290
                                01A8   291 ;++
                                01A8   292 ; FUNCTIONAL DESCRIPTION:
                                01A8   293 ;
                                01A8   294 ;       THIS ROUTINE IS CALLED TO OUTPUT CREF INFORMATION FOR A
                                01A8   295 ;       DIRECTIVE.
                                01A8   296 ;
                                01A8   297 ; INPUTS:
                                01A8   298 ;
                                01A8   299 ;       R1      POINTER TO DIRECTIVE SYMBOL BLOCK
                                01A8   300 ;
                                01A8   301 ;--
                                01A8   302
                                01A8   303 MAC$CREF_DIR::
      4D 0000'CF    00    E1    01A8   304          BBC     #CRF$V_DIR,W^MAC$GL_CRF_FLG,30$ ;BRANCH IF NO CREF OF DIRECTIVES
         49 6B    1F    E0    01AE   305          BBS     #FLG$V_XCRF,(R11),30$   ;BRANCH IF .NOCROSS IN EFFECT
                                01B2   306 ;
                                01B2   307 ; THE DIRECTIVE TABLE, LIKE THE OPCODE TABLE, IS IN READ-ONLY MEMORY.
                                01B2   308 ; THUS, WE MUST ALWAYS DO AN INSERT KEY.
                                01B2   309 ;
                  51    DD    01B2   310          PUSHL   R1                      ;SAVE R1 (PTR TO SYM BLK)
                  00    DD    01B4   311          PUSHL   #0                      ;STACK FLAGS
        00000000'EF     9F    01B6   312          PUSHAB  L^MAC$GK_ZERO           ;STACK VALUE
           50    04 A1    9A    01BC   313          MOVZBL  SYM$B_NAME(R1),R0       ; Get offset to counted name
      7E   51    50    C3    01C0   314          SUBL3   R0,R1,-(SP)             ; and form its address on stack
              0000'CF     9F    01C4   315          PUSHAB  W^MAC$AL_CRFDRCTB       ;STACK CONTROL TABLE ADDRESS
  00000000'GF     04    FB    01C8   316          CALLS   #4,G^CRF$INSRTKEY       ;INSERT THE KEY
              51    6E    D0    01CF   317          MOVL    (SP),R1                 ;REGET SYMBOL BLOCK ADDRESS
        00000000'8F     DD    01D2   318          PUSHL   #CRF$K_REF              ;THIS IS A REFERENCE
                  00    DD    01D8   319          PUSHL   #0                      ;REFERENCER FLAGS
              0000'CF     9F    01DA   320          PUSHAB  W^MAC$AB_LPBUF          ;REFERENCER NAME ADDRESS
           50    04 A1    9A    01DE   321          MOVZBL  SYM$B_NAME(R1),R0       ; Get offset to counted name
      7E   51    50    C3    01E2   322          SUBL3   R0,R1,-(SP)             ; and form its address on stack
              0000'CF     9F    01E6   323          PUSHAB  W^MAC$AL_CRFDRCTB       ;STACK CONTROL TABLE ADDRESS
                 FE13     30    01EA   324          BSBW    MAC$CVT_LIN_PAG         ;CONVERT LINE/PAGE
  00000000'GF     05    FB    01ED   325          CALLS   #5,G^CRF$INSRTREF       ;INSERT REFERENCE
              0000'CF     D6    01F4   326          INCL    W^MAC$GL_CRF_DCNT       ;COUNT DIRECTIVE REFERENCE
                  51 8ED0    01F8   327          POPL    R1                      ;GET SYMBOL BLOCK ADDRESS BACK
                     05    01FB   328 30$:       RSB
                           01FC   329
                           01FC   330          .END
```

```
$$.TMP1                         = 00000002
$$.TMP2                         = 00000060
AB                              = 00000001
AD                              = 0000C008
AF                              = 00008004
AG                              = 0000A008
AH                              = 00009010
AL                              = 00000004
AO                              = 00000010
AQ                              = 00000008
AW                              = 00000002
B                               = 00000001
BIT...                          = 00000005
CRF$INSRTKEY                    ******** X   03
CRF$INSRTREF                    ******** X   03
CRF$K_REF                       ******** X   03
CRF$M_DEFAULT                   = 00000012
CRF$M_DIR                       = 00000001
CRF$M_MACROS                    = 00000002
CRF$M_OPCODES                   = 00000004
CRF$M_REGISTERS                 = 00000008
CRF$M_SYMBOLS                   = 00000010
CRF$V_DIR                       = 00000000
CRF$V_MACROS                    = 00000001
CRF$V_OPCODES                   = 00000002
CRF$V_REGISTERS                 = 00000003
CRF$V_SYMBOLS                   = 00000004
D                               = 0000C008
F                               = 00008004
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_NTYPEPC                   = 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$V_NTYPEPC                   = 00000025
FLG$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$V_XCRF                      = 0000001F
G                               = 0000A008
H                               = 00009010
L                               = 00000004
MAB$B_ARGNO                       00000005
MAB$B_NAME                        00000004
MAB$K_BLKSIZ                      0000000C
MAB$L_DVPTR                       00000008
MAB$L_LINK                        00000000
MAB$W_DVLEN                       00000006
MAC$AB_LPBUF                      ********    X   03
MAC$AL_CRFDRCTB                   ********    X   03
MAC$AL_CRFMCCTB                   ********    X   03
MAC$AL_CRFOPCTB                   ********    X   03
MAC$AL_CRFSYCTB                   ********    X   03
MAC$CREF_DIR                      000001A8 RG     03
MAC$CREF_MACRO                    00000113 RG     03
MAC$CREF_OPCODE                   0000015F RG     03
MAC$CREF_SYM                      000000A9 RG     03
MAC$CRF_ERROR                     0000004F RG     03
MAC$CRF_OUTPUT                    0000006F RG     03
MAC$CVT_LIN_PAG                   00000000 RG     03
MAC$DEC_OUT_L2X                   ********      X 03
MAC$ERR_PUT                       ********      X 03
MAC$GK_ZERO                       ********      X 03
MAC$GL_CRF_CNT                    ********      X 03
MAC$GL_CRF_DCNT                   ********      X 03
MAC$GL_CRF_FLG                    ********      X 03
MAC$GL_CRF_MCNT                   ********      X 03
MAC$GL_CRF_OCNT                   ********      X 03
MAC$GL_FLAGS                      ********      X 03
MAC$GL_LINE_CNT                   ********      X 03
MAC$GL_MOPPTR                     ********      X 03
MAC$GL_SRCPAG                     ********      X 03
MAC$GL_STATUS                     ********      X 03
MAC$GT_SCB                        ********      X 03
MAC$GW_LST_INST                   ********      X 03
MAC$GW_LST_LINE                   ********      X 03
MAC$LAST_CHANCE                   ********      X 03
MAC$LIST_RAB                      ********      X 03
MAC$LST_PAG_HDR                   ********      X 03
MAC$MACRO_NAME                    ********      X 03
MB                              = 00000041
MD                              = 0000C048
MF                              = 00008044
MG                              = 0000A048
MH                              = 00009050
ML                              = 00000044
MNB$B_ARGCT                       00000017
MNB$B_NAME                        00000004
MNB$K_BLKSIZ                      0000001C
MNB$L_ARGP                        00000018
MNB$L_CRSYMF                      00000013
MNB$L_LINK                        00000000
MNB$L_PAGC                        0000000F
MNB$L_PAGP                        0000000B
MNB$L_TXTP                        00000005
MNB$W_FLAG                        00000009
MO                              = 00000050
MQ                              = 00000048
MW                              = 00000042
MXB$K_BLKSIZ                      00000008
MXB$L_LINK                        00000000
MXB$L_PAGES                       00000004
O                               = 00000010
OPD$M_ADDR                      = 00000000
OPD$M_BB                        = 000000A1
OPD$M_BW                        = 000000C2
OPD$M_D_FLOAT                   = 0000C000
OPD$M_FLOAT                     = 00008000
OPD$M_G_FLOAT                   = 0000A000
OPD$M_H_FLOAT                   = 00009000
OPD$M_MODE                      = 000003E0
OPD$M_MODIFY                    = 00000040
OPD$M_NOT_32F                   = 00007000
OPD$M_READ                      = 00000020
OPD$M_VIELD                     = 00000080
OPD$M_WRITE                     = 00000060
OPD$S_MODE                      = 00000005
OPD$S_SIZE                      = 00000005
OPD$V_D_FLOAT                   = 0000000E
OPD$V_FLOAT                     = 0000000F
OPD$V_G_FLOAT                   = 0000000D
OPD$V_H_FLOAT                   = 0000000C
OPD$V_MODE                      = 00000005
OPD$V_SIZE                      = 00000000
OPF$M_LASTOPR                   = 00002000
OPF$M_OPTEXP                    = 00001000
OPF$V_LASTOPR                   = 0000000D
OPF$V_OPTEXP                    = 0000000C
PSC$B_NAME                        00000004
PSC$B_SEG                         0000000C
PSC$B_UNUSED                      0000000B
PSC$K_BLKSIZ                      00000013
PSC$K_NO_OPTNS                  = 0000000A
PSC$L_CURLOC                      0000000F
PSC$L_LINK                        00000000
PSC$L_MAXLGTH                     00000005
```

```
PSC$M_ABS                       = FFFFFFF7
PSC$M_ALIGNFLG                  = 00004000
PSC$M_ALLOPTNS                  = 000003FF
PSC$M_BYTE                      = 00004000
PSC$M_CON                       = FFFFFFFB
PSC$M_DEFAULT                   = 000001C8
PSC$M_EXE                       = 000000C0
PSC$M_GBL                       = 00000010
PSC$M_LCL                       = FFFFFFEF
PSC$M_LIB                       = 00000002
PSC$M_LONG                      = 00004800
PSC$M_NOEXE                     = FFFFFFBF
PSC$M_NOPIC                     = FFFFFFFE
PSC$M_NORD                      = FFFFFF7F
PSC$M_NOSHR                     = FFFFFFDF
PSC$M_NOVEC                     = FFFFFDFF
PSC$M_NOWRT                     = FFFFFEFF
PSC$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$M_USR                       = FFFFFFFD
PSC$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$W_OPTIONS                     0000000D
Q                               = 00000008
RAB$L_RBF                       = 00000028
RAB$W_RSZ                       = 00000022
RB                              = 00000021
RD                              = 0000C028
RF                              = 00008024
RG                              = 0000A028
RH                              = 00009030
RL                              = 00000024
RO                              = 00000030
RQ                              = 00000028
RW                              = 00000022
SIZ...                          = 00000001
SUM_B_FLAGS                       0000001C
SUM_K_BLN                         0000001D
SUM_L_ISDATA                      00000004
SUM_L_STS                         00000000
SUM_M_AUDIT                     = 00000001
SUM_M_AUDITNEW                  = 00000002
SUM_M_DELETE                    = 00000010
SUM_M_SRCUPD                    = 00000004
SUM_M_SUBCLSH                   = 00000008
SUM_Q_AUDDS                       00000008
SUM_Q_FILESP                      00000010
SUM_V_AUDIT                     = 00000000
SUM_V_AUDITNEW                  = 00000001
SUM_V_DELETE                    = 00000004
SUM_V_SRCUPD                    = 00000002
SUM_V_SUBCLSH                   = 00000003
SUM_W_INSERT_NO                   0000001A
SUM_W_LINE_NO                     00000018
SYM$B_NAME                        00000004
SYM$B_SEG                         0000000C
SYM$B_TOKEN                       0000000B
SYM$K_BLKSIZ                      0000000D
SYM$K_MAXLEN                    = 0000001F
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$V_CRFO                      = 0000000D
SYM$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
```

```
SYS$PUT                         ********  GX   03
SYS$PUTMSG                      ********   X   03
VB                              = 00000081
VD                              = 0000C088
VF                              = 00008084
VG                              = 0000A088
VH                              = 00009090
VL                              = 00000084
VO                              = 00000090
VQ                              = 00000088
VW                              = 00000082
W                               = 00000002
WB                              = 00000061
WD                              = 0000C068
WF                              = 00008064
WG                              = 0000A068
WH                              = 00009070
WL                              = 00000064
WO                              = 00000070
WQ                              = 00000068
WW                              = 00000062
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
ABS	0000001D (29.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC$RO_CODE_P15	000001FC (508.)	03 (3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.02	00:00:01.96
Command processing	134	00:00:00.40	00:00:03.10
Pass 1	218	00:00:03.22	00:00:16.16
Symbol table sort	0	00:00:00.41	00:00:00.87
Pass 2	75	00:00:00.76	00:00:03.63
Symbol table output	42	00:00:00.18	00:00:00.34
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	508	00:00:05.01	00:00:26.08

```
The working set limit was 1500 pages.
28512 bytes (56 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 490 non-local and 11 local symbols.
330 source lines were read in Pass 1, producing 17 object records in Pass 2.
22 pages of virtual memory were used to define 17 macros.
```

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

Macro library name                   Macros defined
------------------                   --------------
_$255$DUA28:[SHRLIB]SUM.MLB;1                  3
_$255$DUA28:[MACRO.OBJ]MACRO.MLB;1             5
_$255$DUA28:[SYSLIB]STARLET.MLB;2              7
TOTALS (all libraries)                        15

595 GETS were required to define 15 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:CRFSUB/OBJ=OBJ$:CRFSUB MSRC$:CRFSUB/UPDATE=(ENH$:CRFSUB)+LIB$:MACRO/LIB+SHRLIB$:SUM/LIB
```

ACTPRI LIS

ARGSCN LIS

BDYSCN LIS

CRFSUB LIS

ACTOPC LIS

ACTSTA LIS

APSECT LIS

CRFDAT LIS

ACTREF LIS

COMPUT LIS

DATA
LIS

DEFINE
LIS

FLOAT
LIS

ERRMSG
LIS

GETARG
LIS

DATA1
LIS

INPUT
LIS

ERROR
LIS

FINISH
LIS

GETCMD
LIS