```
*****************************************************************
* Copyright 2008 Automated Software Tools Corporation           *
* This source code is part of z390 assembler/emulator package   *
* The z390 package is distributed under GNU general public license  *
* Author - Don Higgins                                          *
* Date   - 08/13/08                                             *
*****************************************************************
* 08/13/22 RPI 896 TRANSLATE Z390 ZSTRMAC EXTENSIONS TO STD HLASM
*          1.  Z390 BOOTSTRAP VER - RT\TEST\ZSTRMAC1.MLC
*          1.  STRUCTURED VERSION - LINKLIB\ZSTRMAC.ZSM
*          2.  GEN HLASM COMP VER - LINKLIB\ZSTRMAC.MLC VIA ZSTRMAC1
*****************************************************************
* ZSTRMAC READS SYSUT1 SOURCE FILE AND OUTPUTS SYSUT2 SOURCE FILE
* WITH TRANSLATION OF FOLLOWING Z390 ZSTRMAC EXTENSIONS TO STD HLASM:
*  1. AIF (EXP)     >   AIF (NOT(EXP)).AIF_N_B
*                   >   ......
*  2. AELSEIF (EXP) >   AGO .AIF_N_E
*                   > .AIF_B AIF (EXP).AIF_N_B+1
*                   >   ......
*  3. AELSE         >   AGO .AIF_N_E
*                   > .AIF_N_B+1 ANOP
*                   >   ......
*  4. AEND          > .AIF_N_E ANOP
*  5. APM NAME      > &APM_N SETA B
*                   >   AGO .APM_N
*                   > .APM_N_B ANOP
*                   >   ......
*  6. AENTRY NAME   > .APM_N ANOP
*                   >   ......
*  7. AEXIT         >   AGO .APM_N_E    (EXIT NON AIF STRUCURE)
*                   >   ......
*     AEND          >   .APM_N_E AGO (&APM_N).APM_N_1,.APM_N_2,
*                   >                                    .APM_N_B
*  8. AWHILE (EXP)  > .AWH_N_T AIF (NOT(EXP)).AWH_N_E
*                   >   ......
*     AEND          >   AGO .AWH_N_T
*                   > .AWH_N_E ANOP
*                   >   ......
*  9. AUNTIL (EXP)  >   AGO .AUN_N
*                   > .AUN_N_T AIF (EXP).AUN_N_E
*                   > .AUN_N ANOP
*                   >   ......
*     AEND          >   AGO .AUN_N_T
*                   > .AUN_N_E ANOP
*                   >   ......
```

```
* 10. ASELECT (EXP) >   AGO .ASE_N_AGO
* 11. AWHEN V1,V2   > .ASE_N_B1 ANOP  VN=(N,C'?',X'??', OR (V1,V2)
*                   >   ......
*     AWHEN V2      >   AGO .ASE_N_E
*                   > .ASE_N_B2 ANOP
*                   >   ......
*     AELSE         >   AGO .ASE_N_E
*                   > .ASE_N_X ANOP
*                   >   ......
*     AEND          >   AGO .ASE_N_E
*                   > .ASE_N_G AGO (EXP).ASE_N_B1,.ASE_N_X,.ASE_N_B2
*                   >   AGO .ASE_N_X
*                   > .ASE_N_E ANOP
* 12. :label stmt   > place label in label field without the :
*                     and indent the stmt to start at the original :
*
* NOTES:
*  1. THE ORIGINAL BOOTSTRAP VERSION IS IN RT\TEST\ZSTRMAC1.MLC
*     ALONG WITH THE FIRST TEST PROGRAM TESTZSM1.ZSM WHICH IS
*     TRANSLATED TO TESTZSM1.MLC USING ZSTRMAC1.MLC.
*  2. TO RUN TRANSLATOR USING HLASM:
*     A.  REMOVE DDNAME= EXTENSIONS FROM AREAD AND PUNCH
*     B.  PLACE INPUT SOURCE AFTER PROGRAM SOURCE IN SYSIN.
*     C.  CHANGE EOF LOGIC TO CHECK FOR EOF RECORD SUCH AS "END"
*********************************************************************
         MACRO
         ZSTRMAC
         LCLA  &ERRORS        TOTAL ERROR MESSAGES
         LCLA  &AEND_TOT,&AENTRY_TOT,&AEXIT_TOT,&AIF_TOT,&APM_TOT
         LCLA  &ASELECT_TOT,&AUNTIL_TOT,&AWHEN_TOT,&AWHILE_TOT
         LCLC  &TEXT          LINE OF TEXT READ BY READ_TEXT
         LCLB  &EOF           END OF FILE
         LCLA  &LINE          TOTAL INPUT LINES
         LCLB  &GEN_AIF_ERR   SYNTAX ERROR IN GEN_AIF
         LCLB  &FIND_NAME_ERR SYNTAX ERROR FINDING APM/AENTRY NAME
         LCLB  &FIND_PARM_ERR SYNTAX ERROR FINDING FIRST PARM
         LCLB  &FIND_EXP_ERR  SYNTAX ERROR FINDING (..) FOR
AIF/ASELECT
         LCLB  &GET_VALUE_ERR ERROR PARSING DEC, '?', OR X'??'
         LCLA  &LVL           CURRENT LEVEL OF STRUCTURE
         LCLC  &LVL_TYPE(50) TYPE AIF/ASELECT/AENTRY
         LCLA  &LVL_TCNT(50) TYPE INSTANCE COUNTER
         LCLB  &LVL_TEND(50) TYPE END LABEL REQ FOR MULT BLKS
         LCLA  &LVL_BCNT(50) BLOCK COUNTER WITHIN TYPE INSTANCE
         LCLC  &LVL_ASELECT(50) ASELECT COMPUTED AGO STATEMENT
```

```
         LCLA  &LVL_ASELECT_FIRST(50) ASELECT FIRST WHEN VALUE 0-255
         LCLA  &LVL_ASELECT_LAST(50)  ASELECT LAST  WHEN VALUE 0-255
         LCLB  &LVL_AELSE(50)   AELSE BLOCK DEFINED FOR ASELECT
         LCLA  &IS_OP           START OF OPCODE
         LCLA  &IS_OP_END       ENDOF OF OPCODE+1
         LCLA  &IS_EXP          START OF AIF EXP (...)
         LCLA  &APM_INDEX       INDEX TO APM/AENTRY NAME VIA FIND_NAME
         LCLA  &APM_NAME_TOT    TOTAL PERFORMED ROUTINES
         LCLC  &APM_NAME(100)   NAMES OF PERFORMED ROUTINES
         LCLA  &APM_CNT(100)    EXIT COUNT FOR ROUTINES
         LCLB  &APM_DEF(100)    FLAG FOR DUP AND MISSING ERRORS
.*
.* READ SYUT1 AND OUTPUT SYSUT2 WITH STRUCTURED MACRO CODE
.*
.*       APM   READ_REC
&APM_1_READ_REC SETA    1
           AGO   .APM_1_READ_REC
.APM_1_1   ANOP
.*       AWHILE (NOT &EOF)
.AWH_1_T   ANOP
           AIF    (NOT(NOT &EOF)).AWH_1_E
.*             APM PROC_REC
&APM_2_PROC_REC  SETA    1
                 AGO   .APM_2_PROC_REC
.APM_2_1         ANOP
.*             APM READ_REC
&APM_1_READ_REC  SETA    2
                 AGO   .APM_1_READ_REC
.APM_1_2         ANOP
.*       AEND
           AGO   .AWH_1_T
.AWH_1_E   ANOP
&APM_INDEX SETA 1
.*       AWHILE (&APM_INDEX LE &APM_NAME_TOT)
.AWH_2_T   ANOP
           AIF    (NOT(&APM_INDEX LE &APM_NAME_TOT)).AWH_2_E
.*             AIF (NOT &APM_DEF(&APM_INDEX))
                 AIF (NOT(NOT &APM_DEF(&APM_INDEX))).AIF_1_1
&MSG               SETC 'MISSING AENTRY FOR &APM_NAME(&APM_INDEX)'
.*                 APM  ERR_MSG
&APM_3_ERR_MSG       SETA    1
                     AGO   .APM_3_ERR_MSG
.APM_3_1             ANOP
.*             AEND
.AIF_1_1         ANOP
```

```
&APM_INDEX     SETA &APM_INDEX+1
.*        AEND
            AGO   .AWH_2_T
.AWH_2_E    ANOP
          MNOTE 'ZSTRMAC GENERATED LINES = &LINE'
          MNOTE 'ZSTRMAC TOTAL ERRORS    = &ERRORS'
          MNOTE 'ZSTRMAC TOTAL AEND      = &AEND_TOT'
          MNOTE 'ZSTRMAC TOTAL AENTRY    = &AENTRY_TOT'
          MNOTE 'ZSTRMAC TOTAL AEXIT     = &AEXIT_TOT'
          MNOTE 'ZSTRMAC TOTAL AIF       = &AIF_TOT'
          MNOTE 'ZSTRMAC TOTAL APM       = &APM_TOT'
          MNOTE 'ZSTRMAC TOTAL ASELECT   = &ASELECT_TOT'
          MNOTE 'ZSTRMAC TOTAL AWHEN     = &AWHEN_TOT'
          MNOTE 'ZSTRMAC TOTAL AWHILE    = &AWHILE_TOT'
          MNOTE 'ZSTRMAC TOTAL AUNTIL    = &AUNTIL_TOT'
.*
.* READ LOGICAL RECORD INTO &REC WITH TRAILING COMMENTS IF ANY
.*
.*        AENTRY READ_REC
            AGO   .APM_1_SKIP
.APM_1_READ_REC ANOP
.*        APM   READ_TEXT
&APM_4_READ_TEXT SETA    1
            AGO   .APM_4_READ_TEXT
.APM_4_1    ANOP
          ACTR  10000
.*        AIF   (NOT &EOF)
            AIF   (NOT(NOT &EOF)).AIF_2_1
.*              AIF (K'&TEXT GE 72)
                  AIF (NOT(K'&TEXT GE 72)).AIF_3_1
&REC                SETC '&TEXT'(1,71)
.*                  AIF ('&TEXT'(72,1) NE ' ')
                      AIF (NOT('&TEXT'(72,1) NE ' ')).AIF_4_1
.*                      APM  READ_TEXT
&APM_4_READ_TEXT          SETA    2
                          AGO   .APM_4_READ_TEXT
.APM_4_2                  ANOP
.*                      AWHILE (NOT &EOF
X
                                AND K'&TEXT GE 72
X
                                AND '&TEXT'(1,15) EQ (15)' '
X
                                AND '&TEXT'(72,1) NE ' ')
.AWH_3_T                  ANOP
```

```
                                ZSTRMAC.MLC
                           AIF    (NOT(NOT &EOF
X
                                       AND K'&TEXT GE 72
X
                                       AND '&TEXT'(1,15) EQ (15)' '
X
                                       AND '&TEXT'(72,1) NE '
')).AWH_3_E
&REC                               SETC '&REC'.'&TEXT'(16,71-15)
.*                                 APM  READ_TEXT
&APM_4_READ_TEXT                     SETA    3
                                     AGO   .APM_4_READ_TEXT
.APM_4_3                             ANOP
.*                               AEND
                                   AGO   .AWH_3_T
.AWH_3_E                           ANOP
.*                               AIF  (NOT &EOF)
                                   AIF  (NOT(NOT &EOF)).AIF_5_1
.*                                    AIF  (K'&TEXT GE 16
X
                                          AND '&TEXT'(1,15) EQ (15)' ')
                                     AIF  (NOT(K'&TEXT GE 16
X
                                           AND '&TEXT'(1,15) EQ (15)'
')).X
                AIF_6_1
&REC                                  SETC '&REC'.'&TEXT'(16,*)
.*                                 AELSE
                                     AGO   .AIF_6_E
.AIF_6_1                             ANOP
&MSG                                    SETC 'INVALID CONTINUATION'
.*                                      APM  ERR_MSG
&APM_3_ERR_MSG                            SETA    2
                                          AGO   .APM_3_ERR_MSG
.APM_3_2                                  ANOP
.*                                    AEND
.AIF_6_E                                ANOP
.*                               AELSE
                                   AGO   .AIF_5_E
.AIF_5_1                           ANOP
&MSG                                  SETC 'END OF FILE ON CONTINUE'
.*                                    APM  ERR_MSG
&APM_3_ERR_MSG                          SETA    3
                                        AGO   .APM_3_ERR_MSG
.APM_3_3                                ANOP
```

```
.*                    AEND
.AIF_5_E                ANOP
.*                  AEND
.AIF_4_1              ANOP
.*              AELSE
                  AGO   .AIF_3_E
.AIF_3_1          ANOP
&REC                SETC '&TEXT'(1,*)
.*              AEND
.AIF_3_E          ANOP
.*        AEND
.AIF_2_1    ANOP
.*        AEND
            AGO   (&APM_1_READ_REC).APM_1_1,.APM_1_2
.APM_1_SKIP ANOP
.*
.* READ LOGICAL LINE INTO &TEXT AND SET &EOF IF END OF FILE
.*
.*        AENTRY READ_TEXT
            AGO   .APM_4_SKIP
.APM_4_READ_TEXT ANOP
&TEXT     AREAD DDNAME=SYSUT1
.*        AIF ('&TEXT' EQ '')
            AIF (NOT('&TEXT' EQ '')).AIF_7_1
&EOF             SETB 1
.*        AELSE
            AGO   .AIF_7_E
.AIF_7_1    ANOP
&LINE            SETA &LINE+1
.*        AEND
.AIF_7_E    ANOP
.*        AEND
            AGO   (&APM_4_READ_TEXT).APM_4_1,.APM_4_2,.APM_4_3
.APM_4_SKIP ANOP
.*
.* PROCESS REC BY SCANNING FOR A??? OPCODES AND GENERATING
.* COMMENT AND GENERATED CODE ELSE COPY REC
.*
.*        AENTRY PROC_REC
            AGO   .APM_2_SKIP
.APM_2_PROC_REC ANOP
.*        APM   FIND_OPCODE
&APM_5_FIND_OPCODE SETA    1
            AGO   .APM_5_FIND_OPCODE
.APM_5_1    ANOP
```

```
.*       AIF   ('&OPCODE'(1,1) NE 'A')
           AIF   (NOT('&OPCODE'(1,1) NE 'A')).AIF_8_1
.*             APM COPY_REC
&APM_6_COPY_REC  SETA    1
                 AGO   .APM_6_COPY_REC
.APM_6_1         ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AIF')
           AGO   .AIF_8_E
.AIF_8_1   ANOP
           AIF       (NOT('&OPCODE' EQ 'AIF')).AIF_8_2
.*             APM PROC_AIF
&APM_7_PROC_AIF  SETA    1
                 AGO   .APM_7_PROC_AIF
.APM_7_1         ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AELSE')
           AGO   .AIF_8_E
.AIF_8_2   ANOP
           AIF       (NOT('&OPCODE' EQ 'AELSE')).AIF_8_3
.*             APM PROC_AELSE
&APM_8_PROC_AELSE SETA    1
                 AGO   .APM_8_PROC_AELSE
.APM_8_1         ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AELSEIF')
           AGO   .AIF_8_E
.AIF_8_3   ANOP
           AIF       (NOT('&OPCODE' EQ 'AELSEIF')).AIF_8_4
.*             APM PROC_AELSEIF
&APM_9_PROC_AELSEIF SETA    1
                 AGO   .APM_9_PROC_AELSEIF
.APM_9_1         ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AEND')
           AGO   .AIF_8_E
.AIF_8_4   ANOP
           AIF       (NOT('&OPCODE' EQ 'AEND')).AIF_8_5
.*             APM PROC_AEND
&APM_10_PROC_AEND SETA    1
                 AGO   .APM_10_PROC_AEND
.APM_10_1        ANOP
.*       AELSEIF   ('&OPCODE' EQ 'APM')
           AGO   .AIF_8_E
.AIF_8_5   ANOP
           AIF       (NOT('&OPCODE' EQ 'APM')).AIF_8_6
.*             APM PROC_APM
&APM_11_PROC_APM SETA    1
                 AGO   .APM_11_PROC_APM
```

```
.APM_11_1        ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AENTRY')
           AGO   .AIF_8_E
.AIF_8_6   ANOP
           AIF       (NOT('&OPCODE' EQ 'AENTRY')).AIF_8_7
.*             APM PROC_AENTRY
&APM_12_PROC_AENTRY SETA    1
                 AGO   .APM_12_PROC_AENTRY
.APM_12_1        ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AEXIT')
           AGO   .AIF_8_E
.AIF_8_7   ANOP
           AIF       (NOT('&OPCODE' EQ 'AEXIT')).AIF_8_8
.*             APM PROC_AEXIT
&APM_13_PROC_AEXIT SETA    1
                 AGO   .APM_13_PROC_AEXIT
.APM_13_1        ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AWHILE')
           AGO   .AIF_8_E
.AIF_8_8   ANOP
           AIF       (NOT('&OPCODE' EQ 'AWHILE')).AIF_8_9
.*             APM PROC_AWHILE
&APM_14_PROC_AWHILE SETA    1
                 AGO   .APM_14_PROC_AWHILE
.APM_14_1        ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AUNTIL')
           AGO   .AIF_8_E
.AIF_8_9   ANOP
           AIF       (NOT('&OPCODE' EQ 'AUNTIL')).AIF_8_10
.*             APM PROC_AUNTIL
&APM_15_PROC_AUNTIL SETA    1
                 AGO   .APM_15_PROC_AUNTIL
.APM_15_1        ANOP
.*       AELSEIF   ('&OPCODE' EQ 'ASELECT')
           AGO   .AIF_8_E
.AIF_8_10  ANOP
           AIF       (NOT('&OPCODE' EQ 'ASELECT')).AIF_8_11
.*             APM PROC_ASELECT
&APM_16_PROC_ASELECT SETA    1
                 AGO   .APM_16_PROC_ASELECT
.APM_16_1        ANOP
.*       AELSEIF   ('&OPCODE' EQ 'AWHEN')
           AGO   .AIF_8_E
.AIF_8_11  ANOP
           AIF       (NOT('&OPCODE' EQ 'AWHEN')).AIF_8_12
```

```
.*            APM  PROC_AWHEN
&APM_17_PROC_AWHEN SETA    1
                AGO   .APM_17_PROC_AWHEN
.APM_17_1       ANOP
.*       AELSE
           AGO   .AIF_8_E
.AIF_8_12  ANOP
.*            APM  COPY_REC
&APM_6_COPY_REC SETA    2
                AGO   .APM_6_COPY_REC
.APM_6_2        ANOP
.*       AEND
.AIF_8_E   ANOP
.*       AEND
           AGO   (&APM_2_PROC_REC).APM_2_1
.APM_2_SKIP ANOP
.*
.* FIND_OPCODE - SET &OPCODE, &IS_OP, AND &IS_OP_END
.*
.*       AENTRY FIND_OPCODE
           AGO   .APM_5_SKIP
.APM_5_FIND_OPCODE ANOP
&OPCODE  SETC ' '
&IS_OP   SETA 0
&IS_OP_END SETA 0
&I       SETA ('&REC' INDEX ' ')
.*       AIF  (&I GT 0)
           AIF  (NOT(&I GT 0)).AIF_9_1
&J            SETA ('&REC'(&I,*) FIND 'A:')
.*            AIF (&J EQ 0)
                AIF (NOT(&J EQ 0)).AIF_10_1
.*                AEXIT  AENTRY  NOT A???? SO DON'T RETURN OPCODE
                    AGO   .APM_5_E
.*            AELSEIF ('&REC'(1,2) EQ '.*')
                AGO   .AIF_10_E
.AIF_10_1       ANOP
                AIF     (NOT('&REC'(1,2) EQ '.*')).AIF_10_2
.*                AEXIT  AENTRY  NO OPCODE FOR COMMENTS WITH A? EITHER
                    AGO   .APM_5_E
.*            AELSEIF ('&REC'(1,1) EQ '*')
                AGO   .AIF_10_E
.AIF_10_2       ANOP
                AIF     (NOT('&REC'(1,1) EQ '*')).AIF_10_3
.*                AEXIT  AENTRY
                    AGO   .APM_5_E
```

```
.*            AELSEIF ('&REC'(&I,&J-1) NE (&J-&I)' ')
                AGO   .AIF_10_E
.AIF_10_3       ANOP
                AIF     (NOT('&REC'(&I,&J-1) NE (&J-&I)' ')).AIF_10_4
.*                AEXIT  AENTRY
                    AGO   .APM_5_E
.*            AEND
.AIF_10_4       ANOP
.AIF_10_E       ANOP
&I            SETA &I+&J-1
.*            AIF (&I LT K'&REC-1)
                AIF (NOT(&I LT K'&REC-1)).AIF_11_1
&IS_OP            SETA &I
&J                SETA ('&REC'(&I,*) INDEX ' ')
.*                AIF  (&J EQ 0)
                    AIF  (NOT(&J EQ 0)).AIF_12_1
&I                     SETA K'&REC+1
.*                AELSE
                    AGO   .AIF_12_E
.AIF_12_1           ANOP
&I                     SETA &I+&J-1
.*                AEND
.AIF_12_E           ANOP
&OPCODE           SETC (UPPER '&REC'(&IS_OP,&I-&IS_OP))
&IS_OP_END        SETA &I
.*            AEND
.AIF_11_1       ANOP
.*        AEND
.AIF_9_1    ANOP
.*        AEND
.APM_5_E    ANOP
            AGO   (&APM_5_FIND_OPCODE).APM_5_1
.APM_5_SKIP ANOP
.*
.*  COPY UNKNOWN RECORDS WITH :LABEL MOVED TO LABEL FIELD
.*
.*        AENTRY COPY_REC
            AGO   .APM_6_SKIP
.APM_6_COPY_REC ANOP
.*        AIF   (K'&OPCODE GT 1
X
               AND &IS_OP_END LT K'&REC)
            AIF   (NOT(K'&OPCODE GT 1
X
                     AND &IS_OP_END LT K'&REC)).AIF_13_1
```

```
.*              AIF  ('&REC'(&IS_OP,1) EQ ':')
                  AIF  (NOT('&REC'(&IS_OP,1) EQ ':')).AIF_14_1
.*                  APM  FIND_PARM
&APM_18_FIND_PARM    SETA    1
                     AGO   .APM_18_FIND_PARM
.APM_18_1            ANOP
.*                  AIF  (NOT &FIND_PARM_ERR)
                     AIF  (NOT(NOT &FIND_PARM_ERR)).AIF_15_1
&SPACES                 SETA &IS_OP-K'&OPCODE
.*                       AIF (&SPACES LE 0)
                           AIF (NOT(&SPACES LE 0)).AIF_16_1
&SPACES                      SETA 1
.*                       AEND
.AIF_16_1                  ANOP
&REC                     SETC '&REC'(&IS_OP+1,K'&OPCODE-1).(&SPACES)'
'X
                .'&REC'(&IS_PARM,*)
.*                   AEND
.AIF_15_1              ANOP
.*              AEND
.AIF_14_1         ANOP
.*        AEND
.AIF_13_1  ANOP
&PCH_REC SETC '&REC'
.*        APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    1
            AGO   .APM_19_PUNCH_REC
.APM_19_1  ANOP
.*        AEND
            AGO   (&APM_6_COPY_REC).APM_6_1,.APM_6_2
.APM_6_SKIP ANOP
.*
.* AELSE - GEN MACRO COMMENT AND GEN AGO TO AEND AND LABEL FOR ALT.
BLK
.*
.*        AENTRY PROC_AELSE
            AGO   .APM_8_SKIP
.APM_8_PROC_AELSE ANOP
&AELSE_TOT SETA &AELSE_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*        APM   PUNCH_REC
&APM_19_PUNCH_REC SETA    2
            AGO   .APM_19_PUNCH_REC
.APM_19_2  ANOP
.*        AIF   (&LVL GE 1)
```

```
          AIF   (NOT(&LVL GE 1)).AIF_17_1
.*            AIF   (&LVL_TYPE(&LVL) EQ 'AIF')
                AIF   (NOT(&LVL_TYPE(&LVL) EQ 'AIF')).AIF_18_1
.*                  APM PROC_AELSE_AIF
&APM_20_PROC_AELSE_AIF SETA    1
                      AGO   .APM_20_PROC_AELSE_AIF
.APM_20_1             ANOP
.*            AELSEIF   (&LVL_TYPE(&LVL) EQ 'ASELECT')
                AGO   .AIF_18_E
.AIF_18_1       ANOP
                AIF       (NOT(&LVL_TYPE(&LVL) EQ
'ASELECT')).AIF_18_2
.*                  APM PROC_AELSE_ASELECT
&APM_21_PROC_AELSE_ASELECT SETA    1
                      AGO   .APM_21_PROC_AELSE_ASELECT
.APM_21_1             ANOP
.*            AELSE
                AGO   .AIF_18_E
.AIF_18_2       ANOP
&MSG                SETC 'INVALID AELSE TYPE &LVL_TYPE(&LVL)'
.*                  APM ERR_MSG
&APM_3_ERR_MSG        SETA    4
                      AGO   .APM_3_ERR_MSG
.APM_3_4              ANOP
.*            AEND
.AIF_18_E       ANOP
.*       AELSE
           AGO   .AIF_17_E
.AIF_17_1  ANOP
&MSG           SETC 'MISSING AIF OR ASELECT'
.*             APM ERR_MSG
&APM_3_ERR_MSG   SETA    5
                 AGO   .APM_3_ERR_MSG
.APM_3_5         ANOP
.*       AEND
.AIF_17_E  ANOP
.*       AEND
           AGO   (&APM_8_PROC_AELSE).APM_8_1
.APM_8_SKIP ANOP
.*
.* AELSE_AIF
.*
.*       AENTRY PROC_AELSE_AIF
           AGO   .APM_20_SKIP
.APM_20_PROC_AELSE_AIF ANOP
```

```
&LVL_TEND(&LVL) SETB 1  REQUEST AEND TO GEN END TARGET
&PCH_REC SETC (&IS_OP+1)' '.'AGO   .AIF_&LVL_TCNT(&LVL)_E'
.*       APM   PUNCH_REC
&APM_19_PUNCH_REC SETA    3
           AGO   .APM_19_PUNCH_REC
.APM_19_3  ANOP
&PCH_REC SETC '.AIF_&LVL_TCNT(&LVL)_&LVL_BCNT(&LVL)'
.*       APM   PUNCH_LAB
&APM_22_PUNCH_LAB SETA    1
           AGO   .APM_22_PUNCH_LAB
.APM_22_1  ANOP
&LVL_BCNT(&LVL) SETA 0  RESET TO INDICATE NO BLK LABEL REQ
.*       AEND
           AGO   (&APM_20_PROC_AELSE_AIF).APM_20_1
.APM_20_SKIP ANOP
.*
.* AELSE_ASELECT
.*
.*       AENTRY PROC_AELSE_ASELECT
           AGO   .APM_21_SKIP
.APM_21_PROC_AELSE_ASELECT ANOP
.*       AIF   (&LVL_BCNT(&LVL) GT 0)
           AIF   (NOT(&LVL_BCNT(&LVL) GT 0)).AIF_19_1
&PCH_REC       SETC (&IS_OP+1)' '.'AGO   .ASE_&LVL_TCNT(&LVL)_E'
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    4
                 AGO   .APM_19_PUNCH_REC
.APM_19_4        ANOP
.*       AEND
.AIF_19_1  ANOP
&LVL_AELSE(&LVL) SETB 1  INDICATE AELSE BLOCK DEFINED
&PCH_REC SETC '.ASE_&LVL_TCNT(&LVL)_X'
.*       APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    2
           AGO   .APM_22_PUNCH_LAB
.APM_22_2  ANOP
.*       AEND
           AGO   (&APM_21_PROC_AELSE_ASELECT).APM_21_1
.APM_21_SKIP ANOP
.*
.* AELSEIF - GEN MACRO COMMENT AND GEN AIF TO END OF BLK,CUR BLK LAB
.*
.*       AENTRY PROC_AELSEIF
           AGO   .APM_9_SKIP
.APM_9_PROC_AELSEIF ANOP
```

```
&AELSEIF_TOT SETA &AELSEIF_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    5
           AGO   .APM_19_PUNCH_REC
.APM_19_5  ANOP
.*       AIF   (&LVL GE 1)
           AIF   (NOT(&LVL GE 1)).AIF_20_1
.*             AIF  (&LVL_TYPE(&LVL) EQ 'AIF')
                 AIF  (NOT(&LVL_TYPE(&LVL) EQ 'AIF')).AIF_21_1
&LVL_TEND(&LVL)    SETB 1 REQUEST AEND TO GEN END
&PCH_REC           SETC (&IS_OP+1)' '.'AGO   .AIF_&LVL_TCNT(&LVL)_E'
.*                 APM  PUNCH_REC
&APM_19_PUNCH_REC    SETA    6
                     AGO   .APM_19_PUNCH_REC
.APM_19_6            ANOP
&PCH_REC           SETC '.AIF_&LVL_TCNT(&LVL)_&LVL_BCNT(&LVL)'
.*                 APM  PUNCH_LAB
&APM_22_PUNCH_LAB    SETA    3
                     AGO   .APM_22_PUNCH_LAB
.APM_22_3            ANOP
&LVL_BCNT(&LVL)    SETA &LVL_BCNT(&LVL)+1 NEW TARGET
&GEN_AIF_TRUE      SETB 0          GEN BRANCH IF FALSE
&GEN_AIF_TAG       SETC '&LVL_BCNT(&LVL)'
.*                 APM  GEN_AIF
&APM_23_GEN_AIF      SETA    1
                     AGO   .APM_23_GEN_AIF
.APM_23_1            ANOP
.*                 AIF   (&GEN_AIF_ERR)
                     AIF   (NOT(&GEN_AIF_ERR)).AIF_22_1
&MSG                     SETC 'AELSEIF AIF ERROR'
.*                       APM  ERR_MSG
&APM_3_ERR_MSG             SETA    6
                           AGO   .APM_3_ERR_MSG
.APM_3_6                   ANOP
.*                 AELSE
                     AGO   .AIF_22_E
.AIF_22_1            ANOP
.*                       APM  PUNCH_REC
&APM_19_PUNCH_REC          SETA    7
                           AGO   .APM_19_PUNCH_REC
.APM_19_7                  ANOP
.*                 AEND
.AIF_22_E            ANOP
.*             AELSE
```

```
                  AGO   .AIF_21_E
.AIF_21_1         ANOP
&MSG                SETC 'AELSEIF MISSING AIF ERROR'
.*                  APM  ERR_MSG
&APM_3_ERR_MSG        SETA    7
                      AGO   .APM_3_ERR_MSG
.APM_3_7              ANOP
.*              AEND
.AIF_21_E         ANOP
.*        AELSE
            AGO   .AIF_20_E
.AIF_20_1   ANOP
&MSG            SETC 'AELSEIF MISSING AIF ERROR'
.*              APM  ERR_MSG
&APM_3_ERR_MSG  SETA    8
                AGO   .APM_3_ERR_MSG
.APM_3_8        ANOP
.*        AEND
.AIF_20_E   ANOP
.*        AEND
            AGO   (&APM_9_PROC_AELSEIF).APM_9_1
.APM_9_SKIP ANOP
.*
.* AEND - GEN TERMINATION FOR AENTRY,AIF,ASELECT,AUNTIL,AWHILE
.*
.*        AENTRY PROC_AEND
            AGO   .APM_10_SKIP
.APM_10_PROC_AEND ANOP
&AEND_TOT SETA &AEND_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*        APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    8
            AGO   .APM_19_PUNCH_REC
.APM_19_8   ANOP
.*        AIF   (&LVL GE 1)
            AIF   (NOT(&LVL GE 1)).AIF_23_1
.*              AIF   (&LVL_TYPE(&LVL) EQ 'AIF')
                AIF   (NOT(&LVL_TYPE(&LVL) EQ 'AIF')).AIF_24_1
.*                  APM PROC_AEND_AIF
&APM_24_PROC_AEND_AIF  SETA    1
                       AGO   .APM_24_PROC_AEND_AIF
.APM_24_1              ANOP
.*              AELSEIF   (&LVL_TYPE(&LVL) EQ 'AWHILE')
                AGO   .AIF_24_E
.AIF_24_1       ANOP
```

```
                AIF       (NOT(&LVL_TYPE(&LVL) EQ 'AWHILE')).AIF_24_2
.*                  APM PROC_AEND_AWHILE
&APM_25_PROC_AEND_AWHILE SETA    1
                    AGO   .APM_25_PROC_AEND_AWHILE
.APM_25_1           ANOP
.*            AELSEIF   (&LVL_TYPE(&LVL) EQ 'ASELECT')
                AGO   .AIF_24_E
.AIF_24_2       ANOP
                AIF       (NOT(&LVL_TYPE(&LVL) EQ
'ASELECT')).AIF_24_3
.*                  APM PROC_AEND_ASELECT
&APM_26_PROC_AEND_ASELECT SETA    1
                    AGO   .APM_26_PROC_AEND_ASELECT
.APM_26_1           ANOP
.*            AELSEIF   (&LVL_TYPE(&LVL) EQ 'AENTRY')
                AGO   .AIF_24_E
.AIF_24_3       ANOP
                AIF       (NOT(&LVL_TYPE(&LVL) EQ 'AENTRY')).AIF_24_4
.*                  APM PROC_AEND_AENTRY
&APM_27_PROC_AEND_AENTRY SETA    1
                    AGO   .APM_27_PROC_AEND_AENTRY
.APM_27_1           ANOP
.*            AELSEIF   (&LVL_TYPE(&LVL) EQ 'AUNTIL')
                AGO   .AIF_24_E
.AIF_24_4       ANOP
                AIF       (NOT(&LVL_TYPE(&LVL) EQ 'AUNTIL')).AIF_24_5
.*                  APM PROC_AEND_AUNTIL
&APM_28_PROC_AEND_AUNTIL SETA    1
                    AGO   .APM_28_PROC_AEND_AUNTIL
.APM_28_1           ANOP
.*            AELSE
                AGO   .AIF_24_E
.AIF_24_5       ANOP
&MSG               SETC 'AEND INVALID TYPE &LVL_TYPE(&LVL)'
.*                 APM  ERR_MSG
&APM_3_ERR_MSG       SETA    9
                     AGO   .APM_3_ERR_MSG
.APM_3_9             ANOP
.*            AEND
.AIF_24_E       ANOP
.*       AELSE
           AGO   .AIF_23_E
.AIF_23_1  ANOP
&MSG           SETC 'AEND MISSING AIF OR OTHER STRUCTURE'
.*             APM  ERR_MSG
```

```
&APM_3_ERR_MSG   SETA    10
                 AGO   .APM_3_ERR_MSG
.APM_3_10        ANOP
.*       AEND
.AIF_23_E  ANOP
.*       AEND
           AGO   (&APM_10_PROC_AEND).APM_10_1
.APM_10_SKIP ANOP
.*
.* AEND_AENTRY
.*
.*       AENTRY PROC_AEND_AENTRY
           AGO   .APM_27_SKIP
.APM_27_PROC_AEND_AENTRY ANOP
&APM_INDEX SETA &LVL_BCNT(&LVL)
.*       AIF   (&APM_CNT(&APM_INDEX) GT 0)
           AIF   (NOT(&APM_CNT(&APM_INDEX) GT 0)).AIF_25_1
.*             AIF  (&LVL_TEND(&LVL))
                 AIF  (NOT(&LVL_TEND(&LVL))).AIF_26_1
&PCH_REC            SETC '.APM_&APM_INDEX._E'
.*                  APM PUNCH_LAB
&APM_22_PUNCH_LAB     SETA    4
                      AGO   .APM_22_PUNCH_LAB
.APM_22_4             ANOP
.*             AEND
.AIF_26_1        ANOP
&PCH_REC       SETC (&IS_OP+1)' '.'AGO
(&&APM_&APM_INDEX._&APM_NAME(&X
               APM_INDEX)).APM_&APM_INDEX._1'
&I             SETA 2
.*             AWHILE (&I LE &APM_CNT(&APM_INDEX))
.AWH_4_T         ANOP
                 AIF    (NOT(&I LE &APM_CNT(&APM_INDEX))).AWH_4_E
&PCH_REC          SETC '&PCH_REC,.APM_&APM_INDEX._&I'
&I                SETA &I+1
.*             AEND
                 AGO   .AWH_4_T
.AWH_4_E         ANOP
.*             APM PUNCH_REC
&APM_19_PUNCH_REC SETA    9
                 AGO   .APM_19_PUNCH_REC
.APM_19_9        ANOP
.*       AELSE
           AGO   .AIF_25_E
.AIF_25_1  ANOP
```

```
&MSG           SETC 'AENTRY &APM_NAME(&APM_INDEX) NOT USED'
.*             APM ERR_MSG
&APM_3_ERR_MSG   SETA    11
                 AGO   .APM_3_ERR_MSG
.APM_3_11        ANOP
.*       AEND
.AIF_25_E  ANOP
&PCH_REC SETC '.APM_&APM_INDEX._SKIP'
.*       APM PUNCH_LAB
&APM_22_PUNCH_LAB SETA    5
           AGO   .APM_22_PUNCH_LAB
.APM_22_5  ANOP
&LVL     SETA  &LVL-1     CURRENT LEVEL
.*       AEND
           AGO   (&APM_27_PROC_AEND_AENTRY).APM_27_1
.APM_27_SKIP ANOP
.*
.* AEND_AIF
.*
.*       AENTRY PROC_AEND_AIF
           AGO   .APM_24_SKIP
.APM_24_PROC_AEND_AIF ANOP
.*       AIF   (&LVL_BCNT(&LVL) GT 0)
           AIF   (NOT(&LVL_BCNT(&LVL) GT 0)).AIF_27_1
&PCH_REC       SETC '.AIF_&LVL_TCNT(&LVL)_&LVL_BCNT(&LVL)'
.*             APM PUNCH_LAB
&APM_22_PUNCH_LAB SETA    6
                  AGO   .APM_22_PUNCH_LAB
.APM_22_6         ANOP
.*       AEND
.AIF_27_1  ANOP
.*       AIF   (&LVL_TEND(&LVL))
           AIF   (NOT(&LVL_TEND(&LVL))).AIF_28_1
&PCH_REC       SETC '.AIF_&LVL_TCNT(&LVL)_E'
.*             APM PUNCH_LAB
&APM_22_PUNCH_LAB SETA    7
                  AGO   .APM_22_PUNCH_LAB
.APM_22_7         ANOP
.*       AEND
.AIF_28_1  ANOP
&LVL     SETA  &LVL-1     CURRENT LEVEL
.*       AEND
           AGO   (&APM_24_PROC_AEND_AIF).APM_24_1
.APM_24_SKIP ANOP
.*
```

```
.* AEND_AUNTIL
.*
.*       AENTRY PROC_AEND_AUNTIL
           AGO   .APM_28_SKIP
.APM_28_PROC_AEND_AUNTIL ANOP
&PCH_REC SETC (&IS_OP+1)' '.'AGO   .AUN_&LVL_TCNT(&LVL)_T'
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    10
           AGO   .APM_19_PUNCH_REC
.APM_19_10 ANOP
&PCH_REC SETC '.AUN_&LVL_TCNT(&LVL)_E'
.*       APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    8
           AGO   .APM_22_PUNCH_LAB
.APM_22_8  ANOP
&LVL     SETA  &LVL-1     CURRENT LEVEL
.*       AEND
           AGO   (&APM_28_PROC_AEND_AUNTIL).APM_28_1
.APM_28_SKIP ANOP
.*
.* AEND_AWHILE
.*
.*       AENTRY PROC_AEND_AWHILE
           AGO   .APM_25_SKIP
.APM_25_PROC_AEND_AWHILE ANOP
&PCH_REC SETC (&IS_OP+1)' '.'AGO   .AWH_&LVL_TCNT(&LVL)_T'
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    11
           AGO   .APM_19_PUNCH_REC
.APM_19_11 ANOP
&PCH_REC SETC '.AWH_&LVL_TCNT(&LVL)_E'
.*       APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    9
           AGO   .APM_22_PUNCH_LAB
.APM_22_9  ANOP
&LVL     SETA  &LVL-1     CURRENT LEVEL
.*       AEND
           AGO   (&APM_25_PROC_AEND_AWHILE).APM_25_1
.APM_25_SKIP ANOP
.*
.* AEND_ASELECT
.*
.*       AENTRY PROC_AEND_ASELECT
           AGO   .APM_26_SKIP
.APM_26_PROC_AEND_ASELECT ANOP
```

```
.*       AIF   (&LVL_BCNT(&LVL) GT 0)
           AIF   (NOT(&LVL_BCNT(&LVL) GT 0)).AIF_29_1
&PCH_REC       SETC (&IS_OP+1)' '.'AGO   .ASE_&LVL_TCNT(&LVL)_E'
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    12
                 AGO   .APM_19_PUNCH_REC
.APM_19_12       ANOP
&PCH_REC       SETC '.ASE_&LVL_TCNT(&LVL)_G'
.*             APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    10
                 AGO   .APM_22_PUNCH_LAB
.APM_22_10       ANOP
.*             AIF  (&LVL_AELSE(&LVL))
                 AIF  (NOT(&LVL_AELSE(&LVL))).AIF_30_1
&ELSE_LAB           SETC '.ASE_&LVL_TCNT(&LVL)_X'
.*             AELSE
                 AGO   .AIF_30_E
.AIF_30_1        ANOP
&ELSE_LAB           SETC '.ASE_&LVL_TCNT(&LVL)_E'
.*             AEND
.AIF_30_E        ANOP
&PCH_REC       SETC '&LVL_ASELECT(&LVL)'
.*             AIF  (&LVL_ASELECT_FIRST(&LVL) NE 1))
                 AIF  (NOT(&LVL_ASELECT_FIRST(&LVL) NE 1))).AIF_31_1
&OFFSET             SETC '+1-&LVL_ASELECT_FIRST(&LVL)'
&PCH_REC            SETC '&PCH_REC'(1,K'&PCH_REC-1).'&OFFSET)'
.*             AEND
.AIF_31_1        ANOP
&VAL_BLK       SETC  'ASELECT_&LVL_TCNT(&LVL)_VAL_BLK'
&VALUE         SETA  &LVL_ASELECT_FIRST(&LVL)
&COMMA         SETC  ''
.*             AWHILE (&VALUE LE &LVL_ASELECT_LAST(&LVL))
.AWH_5_T         ANOP
                 AIF    (NOT(&VALUE LE
&LVL_ASELECT_LAST(&LVL))).AWH_5_X
               E
.*                     AIF   (&(&VAL_BLK)(&VALUE+1) GT 0)
                         AIF   (NOT(&(&VAL_BLK)(&VALUE+1) GT
0)).AIF_32_X
               1
&PCH_REC                       SETC
'&PCH_REC&COMMA..ASE_&LVL_TCNT(&LVL)_X
               &(&VAL_BLK)(&VALUE+1)'
&COMMA                         SETC  ','
.*                     AELSE
```

```
                        AGO   .AIF_32_E
.AIF_32_1               ANOP
&PCH_REC                    SETC  '&PCH_REC&COMMA&ELSE_LAB'
&COMMA                      SETC  ','
.*                    AEND
.AIF_32_E               ANOP
&VALUE                SETA  &VALUE+1
.*              AEND
                  AGO   .AWH_5_T
.AWH_5_E          ANOP
.*              APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    13
                  AGO   .APM_19_PUNCH_REC
.APM_19_13        ANOP
.*              AIF   (&LVL_AELSE(&LVL))
                  AIF   (NOT(&LVL_AELSE(&LVL))).AIF_33_1
&PCH_REC              SETC (&IS_OP+1)' '.'AGO   .ASE_&LVL_TCNT(&LVL)_X'
.*                    APM  PUNCH_REC
&APM_19_PUNCH_REC       SETA    14
                        AGO   .APM_19_PUNCH_REC
.APM_19_14              ANOP
.*              AEND
.AIF_33_1         ANOP
&PCH_REC        SETC '.ASE_&LVL_TCNT(&LVL)_E'
.*              APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    11
                  AGO   .APM_22_PUNCH_LAB
.APM_22_11        ANOP
&LVL            SETA  &LVL-1     CURRENT LEVEL
.*        AELSE
            AGO   .AIF_29_E
.AIF_29_1  ANOP
&MSG            SETC 'NO WHEN FOUND FOR ASELECT'
.*              APM  ERR_MSG
&APM_3_ERR_MSG   SETA    12
                 AGO   .APM_3_ERR_MSG
.APM_3_12        ANOP
.*        AEND
.AIF_29_E  ANOP
.*        AEND
            AGO   (&APM_26_PROC_AEND_ASELECT).APM_26_1
.APM_26_SKIP ANOP
.*
.* AENTRY - GEN AGO BRANCH AROUND PENTRY/PEND AND LABEL FOR ENTRY
.*
```

```
.*       AENTRY PROC_AENTRY
           AGO   .APM_12_SKIP
.APM_12_PROC_AENTRY ANOP
&AENTRY_TOT SETA &AENTRY_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM   PUNCH_REC
&APM_19_PUNCH_REC SETA    15
           AGO   .APM_19_PUNCH_REC
.APM_19_15 ANOP
.*       APM   FIND_NAME
&APM_29_FIND_NAME SETA    1
           AGO   .APM_29_FIND_NAME
.APM_29_1  ANOP
.*       AIF   (&FIND_NAME_ERR)
           AIF   (NOT(&FIND_NAME_ERR)).AIF_34_1
&MSG           SETC 'AENTRY NAME NOT FOUND'
.*             APM  ERR_MSG
&APM_3_ERR_MSG   SETA    13
                 AGO   .APM_3_ERR_MSG
.APM_3_13        ANOP
.*       AELSEIF (&APM_DEF(&APM_INDEX))
           AGO   .AIF_34_E
.AIF_34_1  ANOP
           AIF     (NOT(&APM_DEF(&APM_INDEX))).AIF_34_2
&MSG           SETC 'AENTRY DUPLICATE NAME FOUND - &NAME'
.*             APM  ERR_MSG
&APM_3_ERR_MSG   SETA    14
                 AGO   .APM_3_ERR_MSG
.APM_3_14        ANOP
.*       AELSE
           AGO   .AIF_34_E
.AIF_34_2  ANOP
&APM_DEF(&APM_INDEX) SETB 1        SET DEFINITION FLAG
&LVL           SETA &LVL+1
&LVL_TYPE(&LVL) SETC 'AENTRY'
&LVL_TEND(&LVL) SETB 0             RESET END LABEL REQ.
&LVL_TCNT(&LVL) SETA &AENTRY_TOT
&LVL_BCNT(&LVL) SETA &APM_INDEX    SAVE FOR AEND
&PCH_REC       SETC (&IS_OP+1)' '.'AGO   .APM_&APM_INDEX._SKIP'
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    16
                 AGO   .APM_19_PUNCH_REC
.APM_19_16       ANOP
&PCH_REC       SETC '.APM_&APM_INDEX._&APM_NAME(&APM_INDEX)'
.*             APM  PUNCH_LAB
```

```
&APM_22_PUNCH_LAB SETA    12
                  AGO   .APM_22_PUNCH_LAB
.APM_22_12        ANOP
.*        AEND
.AIF_34_E  ANOP
.*        AEND
            AGO   (&APM_12_PROC_AENTRY).APM_12_1
.APM_12_SKIP ANOP
.*
.* AEXIT - EXIT TO FIRST MATCHING TYPE FOUND
.*
.*        AENTRY PROC_AEXIT
            AGO   .APM_13_SKIP
.APM_13_PROC_AEXIT ANOP
&AEXIT_TOT SETA &AEXIT_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*        APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    17
            AGO   .APM_19_PUNCH_REC
.APM_19_17 ANOP
.*        APM  FIND_PARM
&APM_18_FIND_PARM SETA    2
            AGO   .APM_18_FIND_PARM
.APM_18_2  ANOP
.*        AIF  (&FIND_PARM_ERR)
            AIF  (NOT(&FIND_PARM_ERR)).AIF_35_1
&MSG           SETC 'AEXIT TYPE PARM NOT FOUND'
.*             APM ERR_MSG
&APM_3_ERR_MSG  SETA    15
                AGO   .APM_3_ERR_MSG
.APM_3_15       ANOP
.*             AEXIT AENTRY
                AGO   .APM_13_E
.*        AEND
.AIF_35_1  ANOP
&EXIT_LVL SETA 0
&TEST_LVL SETA &LVL
.*        AWHILE     (&TEST_LVL GT 0)
.AWH_6_T   ANOP
            AIF        (NOT(&TEST_LVL GT 0)).AWH_6_E
.*              AIF  (&LVL_TYPE(&TEST_LVL) EQ '&PARM')
                AIF  (NOT(&LVL_TYPE(&TEST_LVL) EQ '&PARM')).AIF_36_1
&EXIT_LVL          SETA &TEST_LVL
&TEST_LVL          SETA 0
.*             AELSE
```

```
                  AGO   .AIF_36_E
.AIF_36_1         ANOP
&TEST_LVL            SETA &TEST_LVL-1
.*              AEND
.AIF_36_E         ANOP
.*         AEND
             AGO   .AWH_6_T
.AWH_6_E   ANOP
.*         AIF   (&EXIT_LVL GT 0)
             AIF   (NOT(&EXIT_LVL GT 0)).AIF_37_1
&LVL_TEND(&EXIT_LVL) SETB 1   REQUEST END LABEL
.*              AIF  (&LVL_TYPE(&EXIT_LVL) EQ 'AENTRY')
                  AIF  (NOT(&LVL_TYPE(&EXIT_LVL) EQ 'AENTRY')).AIF_38_1
&APM_INDEX           SETA &LVL_BCNT(&EXIT_LVL)
&PCH_REC             SETC (&IS_OP+1)' '.'AGO   .APM_&APM_INDEX._E'
.*                   APM  PUNCH_REC
&APM_19_PUNCH_REC      SETA    18
                       AGO   .APM_19_PUNCH_REC
.APM_19_18             ANOP
.*              AELSE
                  AGO   .AIF_38_E
.AIF_38_1         ANOP
&PCH_REC             SETC (&IS_OP+1)' '.'AGO
.'.'&LVL_TYPE(&EXIT_LVL)'X
                (1,3).'_&LVL_TCNT(&EXIT_LVL)_E'
.*                   APM  PUNCH_REC
&APM_19_PUNCH_REC      SETA    19
                       AGO   .APM_19_PUNCH_REC
.APM_19_19             ANOP
.*              AEND
.AIF_38_E         ANOP
.*         AELSE
             AGO   .AIF_37_E
.AIF_37_1  ANOP
&MSG             SETC 'AEXIT NOT WITHIN AENTRY, AWHILE, ASELECT'
.*               APM  ERR_MSG
&APM_3_ERR_MSG   SETA    16
                 AGO   .APM_3_ERR_MSG
.APM_3_16        ANOP
.*         AEND
.AIF_37_E  ANOP
.*         AEND
.APM_13_E  ANOP
             AGO   (&APM_13_PROC_AEXIT).APM_13_1
.APM_13_SKIP ANOP
```

```
.*
.* AIF - GEN MACRO COMMENT AND AIF TO GENERATED END LABEL AT NEXT
LEVEL
.*
.*       AENTRY PROC_AIF
           AGO   .APM_7_SKIP
.APM_7_PROC_AIF ANOP
&AIF_TOT SETA  &AIF_TOT+1     AIF COUNTER
&LVL     SETA  &LVL+1     CURRENT LEVEL
&LVL_TYPE(&LVL) SETC 'AIF' CURRENT LEVEL TYPE
&LVL_TCNT(&LVL) SETA &AIF_TOT PRIMARY TYPE COUNTER
&LVL_TEND(&LVL) SETB 0        RESET REQ FOR AELSEIF END LABEL
&LVL_BCNT(&LVL) SETA 1        BLOCK COUNTER (ELSEIF, WHEN)
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    20
           AGO   .APM_19_PUNCH_REC
.APM_19_20 ANOP
&GEN_AIF_TRUE SETB 0                  GEN BRANCH IF FALSE
&GEN_AIF_TAG SETC '&LVL_BCNT(&LVL)'
.*       APM  GEN_AIF
&APM_23_GEN_AIF SETA    2
           AGO   .APM_23_GEN_AIF
.APM_23_2  ANOP
.*       AIF   (&GEN_AIF_ERR)
           AIF   (NOT(&GEN_AIF_ERR)).AIF_39_1
&MSG           SETC 'AIF EXPRESSION SYNTAX ERROR'
.*             APM  ERR_MSG
&APM_3_ERR_MSG   SETA    17
                 AGO   .APM_3_ERR_MSG
.APM_3_17        ANOP
.*       AELSE
           AGO   .AIF_39_E
.AIF_39_1  ANOP
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    21
                 AGO   .APM_19_PUNCH_REC
.APM_19_21       ANOP
.*       AEND
.AIF_39_E  ANOP
.*       AEND
           AGO   (&APM_7_PROC_AIF).APM_7_1
.APM_7_SKIP ANOP
.*
.* APM - GEN AGO TO PERFORMED ROUTINE
```

```
.*
.*       AENTRY PROC_APM
           AGO   .APM_11_SKIP
.APM_11_PROC_APM ANOP
&APM_TOT SETA &APM_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM   PUNCH_REC
&APM_19_PUNCH_REC SETA    22
           AGO   .APM_19_PUNCH_REC
.APM_19_22 ANOP
.*       APM   FIND_NAME
&APM_29_FIND_NAME SETA    2
           AGO   .APM_29_FIND_NAME
.APM_29_2  ANOP
.*       AIF   (&FIND_NAME_ERR)
           AIF   (NOT(&FIND_NAME_ERR)).AIF_40_1
&MSG           SETC 'APM NAME SYNTAX ERROR'
.*             APM  ERR_MSG
&APM_3_ERR_MSG   SETA    18
                 AGO   .APM_3_ERR_MSG
.APM_3_18        ANOP
.*       AELSE
           AGO   .AIF_40_E
.AIF_40_1  ANOP
&APM_CNT(&APM_INDEX) SETA &APM_CNT(&APM_INDEX)+1
&PCH_REC       SETC '&&APM_&APM_INDEX._&APM_NAME(&APM_INDEX)'
&SPACES        SETA &IS_OP-K'&PCH_REC+1
.*             AIF (&SPACES LE 0)
                 AIF (NOT(&SPACES LE 0)).AIF_41_1
&SPACES            SETA 1
.*             AEND
.AIF_41_1        ANOP
&PCH_REC       SETC '&PCH_REC'.(&SPACES)' '.'SETA
&APM_CNT(&APM_INDEX
               X)'
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    23
                 AGO   .APM_19_PUNCH_REC
.APM_19_23       ANOP
&PCH_REC       SETC (&IS_OP+1)' '.'AGO
.APM_&APM_INDEX._&APM_NAME(&APX
               M_INDEX)'
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    24
                 AGO   .APM_19_PUNCH_REC
```

```
.APM_19_24       ANOP
&PCH_REC       SETC '.APM_&APM_INDEX._&APM_CNT(&APM_INDEX)'
.*             APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    13
                 AGO   .APM_22_PUNCH_LAB
.APM_22_13       ANOP
.*       AEND
.AIF_40_E  ANOP
.*       AEND
           AGO   (&APM_11_PROC_APM).APM_11_1
.APM_11_SKIP ANOP
.*
.* ASELECT - GEN AGO TO .ASELECT_N_AGO AND SAVE AGO EXPRESSION
.*
.*       AENTRY PROC_ASELECT
           AGO   .APM_16_SKIP
.APM_16_PROC_ASELECT ANOP
&ASELECT_TOT SETA  &ASELECT_TOT+1     ASELECT COUNTER
&LVL     SETA  &LVL+1     CURRENT LEVEL
&LVL_TYPE(&LVL) SETC 'ASELECT' CURRENT LEVEL TYPE
&LVL_TCNT(&LVL) SETA &ASELECT_TOT ASELECT INSTANCE
&LVL_BCNT(&LVL) SETA 0 RESET ASELECT AWHEN BLOCKS
&LVL_AELSE(&LVL) SETB 0 ASSUME NO AELSE BLOCK
&VAL_BLK SETC  'ASELECT_&LVL_TCNT(&LVL)_VAL_BLK'
         LCLA &(&VAL_BLK)(256)
&LVL_ASELECT_FIRST(&LVL) SETA 257
&LVL_ASELECT_LAST(&LVL) SETA -1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM   PUNCH_REC
&APM_19_PUNCH_REC SETA    25
           AGO   .APM_19_PUNCH_REC
.APM_19_25 ANOP
.*       APM   FIND_EXP
&APM_30_FIND_EXP SETA    1
           AGO   .APM_30_FIND_EXP
.APM_30_1  ANOP
.*       AIF   (&FIND_EXP_ERR)
           AIF   (NOT(&FIND_EXP_ERR)).AIF_42_1
&MSG           SETC 'ASELECT EXPRESSION ERROR'
.*             APM  ERR_MSG
&APM_3_ERR_MSG   SETA    19
                 AGO   .APM_3_ERR_MSG
.APM_3_19        ANOP
.*       AELSE
           AGO   .AIF_42_E
```

```
.AIF_42_1  ANOP
&LVL_ASELECT(&LVL) SETC (&IS_OP+1)' '.'AGO
'.'&REC'(&IS_EXP,&IS_EXP_EX
               ND-&IS_EXP+1)
&I             SETA 1
.*             AWHILE (&I LE 256)
.AWH_7_T         ANOP
                 AIF    (NOT(&I LE 256)).AWH_7_E
&(&VAL_BLK)(&I)     SETA 0
&I                  SETA &I+1
.*             AEND
                 AGO   .AWH_7_T
.AWH_7_E         ANOP
&PCH_REC       SETC (&IS_OP+1)' '.'AGO   .ASE_&LVL_TCNT(&LVL)_G'
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    26
                 AGO   .APM_19_PUNCH_REC
.APM_19_26       ANOP
.*       AEND
.AIF_42_E  ANOP
.*       AEND
           AGO   (&APM_16_PROC_ASELECT).APM_16_1
.APM_16_SKIP ANOP
.*
.* AUNTIL - GEN AGO TO BLOCK, THEN LABEL TEST AIF TO EXIT
.*
.*       AENTRY PROC_AUNTIL
           AGO   .APM_15_SKIP
.APM_15_PROC_AUNTIL ANOP
&AUNTIL_TOT SETA  &AUNTIL_TOT+1   AUNTIL COUNTER
&LVL     SETA  &LVL+1     CURRENT LEVEL
&LVL_TYPE(&LVL) SETC 'AUNTIL' CURRENT LEVEL TYPE
&LVL_TCNT(&LVL) SETA &AUNTIL_TOT PRIMARY TYPE COUNTER
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    27
           AGO   .APM_19_PUNCH_REC
.APM_19_27 ANOP
&PCH_REC SETC (&IS_OP+1)' '.'AGO   .AUN_&LVL_TCNT(&LVL)'
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    28
           AGO   .APM_19_PUNCH_REC
.APM_19_28 ANOP
&PCH_REC SETC '.AUN_&LVL_TCNT(&LVL)_T'
.*       APM  PUNCH_LAB
```

```
&APM_22_PUNCH_LAB SETA    14
           AGO   .APM_22_PUNCH_LAB
.APM_22_14 ANOP
&GEN_AIF_TRUE SETB 1                GEN BRANCH IF TRUE
&GEN_AIF_TAG SETC 'E'
.*       APM  GEN_AIF
&APM_23_GEN_AIF SETA    3
           AGO   .APM_23_GEN_AIF
.APM_23_3  ANOP
.*       AIF   (&GEN_AIF_ERR)
           AIF   (NOT(&GEN_AIF_ERR)).AIF_43_1
&MSG           SETC 'AUNTIL EXPRESSION ERROR'
.*             APM  ERR_MSG
&APM_3_ERR_MSG   SETA    20
                 AGO   .APM_3_ERR_MSG
.APM_3_20        ANOP
.*       AELSE
           AGO   .AIF_43_E
.AIF_43_1  ANOP
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    29
                  AGO   .APM_19_PUNCH_REC
.APM_19_29        ANOP
.*       AEND
.AIF_43_E  ANOP
&PCH_REC SETC '.AUN_&LVL_TCNT(&LVL)'
.*       APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    15
           AGO   .APM_22_PUNCH_LAB
.APM_22_15 ANOP
.*       AEND
           AGO   (&APM_15_PROC_AUNTIL).APM_15_1
.APM_15_SKIP ANOP
.*
.* AWHEN - GEN .ASELECT_N_I LABEL FOR INDEX AND UPDATE INDEX VAL_BLK
.*
.*       AENTRY PROC_AWHEN
           AGO   .APM_17_SKIP
.APM_17_PROC_AWHEN ANOP
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    30
           AGO   .APM_19_PUNCH_REC
.APM_19_30 ANOP
&AWHEN_TOT SETA &AWHEN_TOT+1
```

```
&VAL_BLK SETC  'ASELECT_&LVL_TCNT(&LVL)_VAL_BLK'
.*       AIF   (&LVL GE 1)
           AIF   (NOT(&LVL GE 1)).AIF_44_1
.*             AIF  (&LVL_TYPE(&LVL) EQ 'ASELECT')
                 AIF  (NOT(&LVL_TYPE(&LVL) EQ 'ASELECT')).AIF_45_1
.*                  AIF   (&LVL_BCNT(&LVL) GT 0 OR &LVL_AELSE(&LVL))
                      AIF   (NOT(&LVL_BCNT(&LVL) GT 0 OR
&LVL_AELSE(&LVX
               L))).AIF_46_1
&PCH_REC                   SETC (&IS_OP+1)' '.'AGO
.ASE_&LVL_TCNT(&LVLX
               )_E'
.*                         APM  PUNCH_REC
&APM_19_PUNCH_REC            SETA    31
                             AGO   .APM_19_PUNCH_REC
.APM_19_31                   ANOP
.*                  AEND
.AIF_46_1             ANOP
&LVL_BCNT(&LVL)     SETA &LVL_BCNT(&LVL)+1
.*                  APM  FIND_PARM
&APM_18_FIND_PARM     SETA    3
                      AGO   .APM_18_FIND_PARM
.APM_18_3             ANOP
.*                  AIF   (&FIND_PARM_ERR)
                      AIF   (NOT(&FIND_PARM_ERR)).AIF_47_1
&MSG                     SETC 'AWHEN VALUE ERROR'
.*                       APM ERR_MSG
&APM_3_ERR_MSG             SETA    21
                           AGO   .APM_3_ERR_MSG
.APM_3_21                  ANOP
.*                  AELSE
                      AGO   .AIF_47_E
.AIF_47_1             ANOP
.*                       APM PROC_AWHEN_VALUES
&APM_31_PROC_AWHEN_VALUES  SETA    1
                           AGO   .APM_31_PROC_AWHEN_VALUES
.APM_31_1                  ANOP
.*                  AEND
.AIF_47_E             ANOP
&PCH_REC            SETC '.ASE_&LVL_TCNT(&LVL)_&LVL_BCNT(&LVL)'
.*                  APM  PUNCH_LAB
&APM_22_PUNCH_LAB     SETA    16
                      AGO   .APM_22_PUNCH_LAB
.APM_22_16            ANOP
.*             AELSE
```

```
                  AGO   .AIF_45_E
.AIF_45_1         ANOP
&MSG                 SETC 'AWHEN MISSING ASELECT'
.*                   APM ERR_MSG
&APM_3_ERR_MSG         SETA    22
                       AGO   .APM_3_ERR_MSG
.APM_3_22              ANOP
.*              AEND
.AIF_45_E         ANOP
.*        AELSE
            AGO   .AIF_44_E
.AIF_44_1  ANOP
&MSG            SETC 'AWHEN MISSING ASELECT'
.*              APM ERR_MSG
&APM_3_ERR_MSG  SETA    23
                AGO   .APM_3_ERR_MSG
.APM_3_23       ANOP
.*        AEND
.AIF_44_E  ANOP
.*        AEND
            AGO   (&APM_17_PROC_AWHEN).APM_17_1
.APM_17_SKIP ANOP
.*
.* PROC_WHEN_VALUES V1,V2,(V3,V4) WHERE VN = DEC, C'?', OR X'??'
.*
.*        AENTRY PROC_AWHEN_VALUES
            AGO   .APM_31_SKIP
.APM_31_PROC_AWHEN_VALUES ANOP
&VALUE_CNT SETA 0
.*        AWHILE (&IS_PARM LE K'&REC)
.AWH_8_T   ANOP
           AIF    (NOT(&IS_PARM LE K'&REC)).AWH_8_E
.*             ASELECT (C2A('&REC'(&IS_PARM,1)))
                 AGO   .ASE_1_G
.*                  AWHEN C'('  SET RANGE (V1,V2)
.ASE_1_1              ANOP
&IS_PARM                SETA &IS_PARM+1
.*                      APM GET_VALUE
&APM_32_GET_VALUE         SETA    1
                          AGO   .APM_32_GET_VALUE
.APM_32_1                 ANOP
.*                      AIF  (&GET_VALUE_ERR)
                          AIF  (NOT(&GET_VALUE_ERR)).AIF_48_1
&MSG                         SETC 'INVALID RANGE VALUE'
.*                           APM  ERR_MSG
```

```
                                ZSTRMAC.MLC
&APM_3_ERR_MSG                    SETA    24
                                  AGO   .APM_3_ERR_MSG
.APM_3_24                         ANOP
.*                               AEXIT AENTRY     EXIT AFTER VALUE ERROR
                                  AGO   .APM_31_E
.*                              AEND
.AIF_48_1                         ANOP
&VALUE1                         SETA &VALUE
.*                              AIF ('&REC'(&IS_PARM,1) NE ',')
                                  AIF (NOT('&REC'(&IS_PARM,1) NE
',')).AIF_49_1
&MSG                                 SETC 'MISSING RANGE ,'
.*                                   APM ERR_MSG
&APM_3_ERR_MSG                    SETA    25
                                  AGO   .APM_3_ERR_MSG
.APM_3_25                         ANOP
.*                                   AEXIT AENTRY
                                  AGO   .APM_31_E
.*                              AEND
.AIF_49_1                         ANOP
&IS_PARM                        SETA &IS_PARM+1
.*                              APM GET_VALUE
&APM_32_GET_VALUE                 SETA    2
                                  AGO   .APM_32_GET_VALUE
.APM_32_2                         ANOP
.*                              AIF  (&GET_VALUE_ERR)
                                  AIF  (NOT(&GET_VALUE_ERR)).AIF_50_1
&MSG                                 SETC 'INVALID RANGE VALUE'
.*                                   APM  ERR_MSG
&APM_3_ERR_MSG                    SETA    26
                                  AGO   .APM_3_ERR_MSG
.APM_3_26                         ANOP
.*                                   AEXIT AENTRY  EXIT AFTER VALUE ERROR
                                  AGO   .APM_31_E
.*                              AEND
.AIF_50_1                         ANOP
&VALUE2                         SETA &VALUE
.*                              AIF ('&REC'(&IS_PARM,1) NE ')')
                                  AIF (NOT('&REC'(&IS_PARM,1) NE
')')).AIF_51_1
&MSG                                SETC 'MISSING RANGE )'
.*                                  APM ERR_MSG
&APM_3_ERR_MSG                    SETA    27
                                  AGO   .APM_3_ERR_MSG
.APM_3_27                         ANOP
```

```
.*                          AEXIT AENTRY
                              AGO   .APM_31_E
.*                        AEND
.AIF_51_1                   ANOP
&IS_PARM                  SETA &IS_PARM+1
&VALUE                    SETA &VALUE1
.*                        AWHILE (&VALUE LE &VALUE2)
.AWH_9_T                    ANOP
                            AIF    (NOT(&VALUE LE &VALUE2)).AWH_9_E
.*                            APM SET_VAL_BLK
&APM_33_SET_VAL_BLK             SETA    1
                                AGO   .APM_33_SET_VAL_BLK
.APM_33_1                       ANOP
&(&VAL_BLK)(&VALUE+1)         SETA &LVL_BCNT(&LVL)
&VALUE                        SETA &VALUE+1
.*                        AEND
                            AGO   .AWH_9_T
.AWH_9_E                    ANOP
.*                    AWHEN C' '
                        AGO   .ASE_1_E
.ASE_1_2                ANOP
.*                        AEXIT AWHILE
                            AGO   .AWH_8_E
.*                    AWHEN C','
                        AGO   .ASE_1_E
.ASE_1_3                ANOP
&IS_PARM                  SETA &IS_PARM+1
.*                    AELSE
                        AGO   .ASE_1_E
.ASE_1_X                ANOP
.*                        APM  GET_VALUE
&APM_32_GET_VALUE           SETA    3
                            AGO   .APM_32_GET_VALUE
.APM_32_3                   ANOP
.*                        AIF  (&GET_VALUE_ERR)
                            AIF  (NOT(&GET_VALUE_ERR)).AIF_52_1
&MSG                           SETC 'INVALID VALUE'
.*                             APM  ERR_MSG
&APM_3_ERR_MSG                   SETA    28
                                 AGO   .APM_3_ERR_MSG
.APM_3_28                        ANOP
.*                             AEXIT AENTRY
                                 AGO   .APM_31_E
.*                        AEND
.AIF_52_1                   ANOP
```

```
.*                        APM SET_VAL_BLK
&APM_33_SET_VAL_BLK         SETA    2
                            AGO   .APM_33_SET_VAL_BLK
.APM_33_2                   ANOP
.*              AEND
                  AGO   .ASE_1_E
.ASE_1_G          ANOP
                  AGO
(C2A('&REC'(&IS_PARM,1))+1-64).ASE_1_2,.ASE_1_X,X

.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.AX

SE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_1,.ASEX

_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1X

_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_XX

,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.X

ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASX
                E_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_X,.ASE_1_3
                  AGO   .ASE_1_X
.ASE_1_E          ANOP
.*        AEND
            AGO   .AWH_8_T
.AWH_8_E    ANOP
.*        AIF   (&VALUE_CNT EQ 0)
            AIF   (NOT(&VALUE_CNT EQ 0)).AIF_53_1
&MSG            SETC 'NO AWHEN VALUES FOUND'
.*              APM ERR_MSG
&APM_3_ERR_MSG  SETA    29
                AGO   .APM_3_ERR_MSG
.APM_3_29       ANOP
.*        AEND
.AIF_53_1   ANOP
.*        AEND
.APM_31_E   ANOP
            AGO   (&APM_31_PROC_AWHEN_VALUES).APM_31_1
.APM_31_SKIP ANOP
.*
.* SET_VAL_BLK  AWHEN BLOCK NUMBER FOR VALUE
.*
.*        AENTRY SET_VAL_BLK
            AGO   .APM_33_SKIP
```

```
.APM_33_SET_VAL_BLK ANOP
.*       AIF   (&VALUE LT &LVL_ASELECT_FIRST(&LVL))
           AIF   (NOT(&VALUE LT &LVL_ASELECT_FIRST(&LVL))).AIF_54_1
&LVL_ASELECT_FIRST(&LVL) SETA &VALUE
.*       AEND
.AIF_54_1  ANOP
.*       AIF   (&VALUE GT &LVL_ASELECT_LAST(&LVL))
           AIF   (NOT(&VALUE GT &LVL_ASELECT_LAST(&LVL))).AIF_55_1
&LVL_ASELECT_LAST(&LVL) SETA &VALUE
.*       AEND
.AIF_55_1  ANOP
&INDEX   SETA  &VALUE+1
.*       AIF   (&(&VAL_BLK)(&INDEX) NE 0)
           AIF   (NOT(&(&VAL_BLK)(&INDEX) NE 0)).AIF_56_1
&MSG           SETC 'DUPLICATE AWHEN VALUE &VALUE'
.*             APM ERR_MSG
&APM_3_ERR_MSG   SETA    30
                 AGO   .APM_3_ERR_MSG
.APM_3_30        ANOP
.*       AEND
.AIF_56_1  ANOP
&(&VAL_BLK)(&INDEX) SETA &LVL_BCNT(&LVL) SET BLK # FOR VAL
.*       AEND
           AGO   (&APM_33_SET_VAL_BLK).APM_33_1,.APM_33_2
.APM_33_SKIP ANOP
.*
.* GET_VALUE - DEC, C'?', OR X'??'
.*
.*       AENTRY GET_VALUE
           AGO   .APM_32_SKIP
.APM_32_GET_VALUE ANOP
&GET_VALUE_ERR SETB 0
&VALUE_SET SETB 0
.*       AIF   ('&REC'(&IS_PARM,1) GE '0')
           AIF   (NOT('&REC'(&IS_PARM,1) GE '0')).AIF_57_1
&VALUE         SETA  0
&VALUE_EOF     SETB 0
.*             AWHILE (&IS_PARM LE K'&REC)
.AWH_10_T        ANOP
                 AIF    (NOT(&IS_PARM LE K'&REC)).AWH_10_E
.*                 AIF ('&REC'(&IS_PARM,1) GE '0'
X
                    AND '&REC'(&IS_PARM,1) LE '9')
                     AIF (NOT('&REC'(&IS_PARM,1) GE '0'
X
```

```
                          AND '&REC'(&IS_PARM,1) LE '9')).AIF_58_1
&VALUE_SET              SETB 1
&DIGIT                  SETA '&REC'(&IS_PARM,1)
&VALUE                  SETA &VALUE*10+&DIGIT
&IS_PARM                SETA &IS_PARM+1
.*                    AELSE
                        AGO   .AIF_58_E
.AIF_58_1               ANOP
.*                        AEXIT AWHILE
                            AGO   .AWH_10_E
.*                    AEND
.AIF_58_E               ANOP
.*                AEND
                    AGO   .AWH_10_T
.AWH_10_E           ANOP
.*          AELSEIF ('&REC'(&IS_PARM,1) EQ 'C')
              AGO   .AIF_57_E
.AIF_57_1   ANOP
              AIF     (NOT('&REC'(&IS_PARM,1) EQ 'C')).AIF_57_2
.*                AIF  (&IS_PARM+3 LE K'&REC)
                    AIF  (NOT(&IS_PARM+3 LE K'&REC)).AIF_59_1
.*                      AIF ('&REC'(&IS_PARM+1,1) EQ ''''
X
                         AND '&REC'(&IS_PARM+3,1) EQ '''')
                          AIF (NOT('&REC'(&IS_PARM+1,1) EQ ''''
X
                               AND '&REC'(&IS_PARM+3,1) EQ
''''')).AIF_60_1
&VALUE                    SETA  C2A('&REC'(&IS_PARM+2,1))
&IS_PARM                  SETA &IS_PARM+4  SKIP C'?'
&VALUE_SET                SETB 1
.*                      AELSE
                          AGO   .AIF_60_E
.AIF_60_1                 ANOP
&GET_VALUE_ERR              SETB 1
.*                      AEND
.AIF_60_E                 ANOP
.*                AELSE
                    AGO   .AIF_59_E
.AIF_59_1           ANOP
&GET_VALUE_ERR          SETB 1
.*                AEND
.AIF_59_E           ANOP
.*          AELSEIF ('&REC'(&IS_PARM,1) EQ 'X')
              AGO   .AIF_57_E
```

```
.AIF_57_2  ANOP
           AIF     (NOT('&REC'(&IS_PARM,1) EQ 'X')).AIF_57_3
.*             AIF   (&IS_PARM+4 LE K'&REC)
                 AIF   (NOT(&IS_PARM+4 LE K'&REC)).AIF_61_1
.*                   AIF ('&REC'(&IS_PARM+1,1) EQ ''''
X
                      AND '&REC'(&IS_PARM+4,1) EQ '''')
                       AIF (NOT('&REC'(&IS_PARM+1,1) EQ ''''
X
                            AND '&REC'(&IS_PARM+4,1) EQ
''''')).AIF_62_1
&VALUE                   SETA X2A('&REC'(&IS_PARM+2,2))
&IS_PARM                 SETA &IS_PARM+5 SKIP X'??'
&VALUE_SET               SETB 1
.*                     AELSE
                         AGO   .AIF_62_E
.AIF_62_1                ANOP
&GET_VALUE_ERR            SETB 1
.*                     AEND
.AIF_62_E                ANOP
.*             AELSE
                 AGO   .AIF_61_E
.AIF_61_1        ANOP
&GET_VALUE_ERR       SETB 1
.*             AEND
.AIF_61_E        ANOP
.*       AELSE
           AGO   .AIF_57_E
.AIF_57_3  ANOP
&GET_VALUE_ERR SETB 1
.*       AEND
.AIF_57_E  ANOP
.*       AIF   (&VALUE_SET)
           AIF   (NOT(&VALUE_SET)).AIF_63_1
&VALUE_CNT     SETA &VALUE_CNT+1
.*             AIF  (&VALUE LT 0 OR &VALUE GT 255)  OUT OF RANGE
                 AIF  (NOT(&VALUE LT 0 OR &VALUE GT 255)).AIF_64_1
OUX
               T OF RANGE
&GET_VALUE_ERR      SETB 1
.*             AEND
.AIF_64_1        ANOP
.*       AELSE
           AGO   .AIF_63_E
.AIF_63_1  ANOP
```

```
&GET_VALUE_ERR SETB 1
.*       AEND
.AIF_63_E  ANOP
.*       AEND
           AGO   (&APM_32_GET_VALUE).APM_32_1,.APM_32_2,.APM_32_3
.APM_32_SKIP ANOP
.*
.* AWHILE - GEN LABELD AIF TO END
.*
.*       AENTRY PROC_AWHILE
           AGO   .APM_14_SKIP
.APM_14_PROC_AWHILE ANOP
&AWHILE_TOT SETA  &AWHILE_TOT+1   AWHILE COUNTER
&LVL     SETA  &LVL+1     CURRENT LEVEL
&LVL_TYPE(&LVL) SETC 'AWHILE' CURRENT LEVEL TYPE
&LVL_TCNT(&LVL) SETA &AWHILE_TOT PRIMARY TYPE COUNTER
&PCH_REC SETC '.*'.'&REC'(3,*)
.*       APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    32
           AGO   .APM_19_PUNCH_REC
.APM_19_32 ANOP
&PCH_REC SETC '.AWH_&LVL_TCNT(&LVL)_T'
.*       APM  PUNCH_LAB
&APM_22_PUNCH_LAB SETA    17
           AGO   .APM_22_PUNCH_LAB
.APM_22_17 ANOP
&GEN_AIF_TRUE SETB 0                GEN BRANCH IF FALSE
&GEN_AIF_TAG SETC 'E'
.*       APM  GEN_AIF
&APM_23_GEN_AIF SETA    4
           AGO   .APM_23_GEN_AIF
.APM_23_4  ANOP
.*       AIF   (&GEN_AIF_ERR)
           AIF   (NOT(&GEN_AIF_ERR)).AIF_65_1
&MSG           SETC 'AWHILE EXPRESSION ERROR'
.*             APM ERR_MSG
&APM_3_ERR_MSG   SETA    31
                 AGO   .APM_3_ERR_MSG
.APM_3_31        ANOP
.*       AELSE
           AGO   .AIF_65_E
.AIF_65_1  ANOP
.*             APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    33
                 AGO   .APM_19_PUNCH_REC
```

```
.APM_19_33       ANOP
.*       AEND
.AIF_65_E  ANOP
.*       AEND
           AGO   (&APM_14_PROC_AWHILE).APM_14_1
.APM_14_SKIP ANOP
.*
.* FIND_NAME OPERAND AND SET APM_INDEX TO EXISTING OR NEW ENTRY
.* SET FIND_NAME_ERR IF PARM ERROR
.*
.*       AENTRY FIND_NAME
           AGO   .APM_29_SKIP
.APM_29_FIND_NAME ANOP
&FIND_NAME_ERR SETB 0
.*       APM   FIND_PARM
&APM_18_FIND_PARM SETA    4
           AGO   .APM_18_FIND_PARM
.APM_18_4  ANOP
.*       AIF   (&FIND_PARM_ERR)
           AIF   (NOT(&FIND_PARM_ERR)).AIF_66_1
&FIND_NAME_ERR SETB 1
.*       AELSE
           AGO   .AIF_66_E
.AIF_66_1  ANOP
&NAME          SETC (UPPER '&PARM')
&APM_INDEX     SETA 1
.*             AWHILE (&APM_INDEX LE &APM_NAME_TOT)
.AWH_11_T        ANOP
                 AIF    (NOT(&APM_INDEX LE &APM_NAME_TOT)).AWH_11_E
.*                  AIF  ('&APM_NAME(&APM_INDEX)' EQ '&NAME')
                      AIF  (NOT('&APM_NAME(&APM_INDEX)' EQ
'&NAME')).AIX
               F_67_1
.*                         AEXIT AENTRY  EXIT WITH APM_INDEX SET
                             AGO   .APM_29_E
.*                  AEND
.AIF_67_1            ANOP
&APM_INDEX           SETA &APM_INDEX+1
.*             AEND
                 AGO   .AWH_11_T
.AWH_11_E        ANOP
.*             AIF  (&APM_INDEX GT &APM_NAME_TOT)
                 AIF  (NOT(&APM_INDEX GT &APM_NAME_TOT)).AIF_68_1
&APM_NAME_TOT       SETA &APM_INDEX
&APM_NAME(&APM_INDEX) SETC '&NAME'
```

```
.*             AEND
.AIF_68_1        ANOP
.*         AEND
.AIF_66_E  ANOP
.*         AEND
.APM_29_E  ANOP
             AGO   (&APM_29_FIND_NAME).APM_29_1,.APM_29_2
.APM_29_SKIP ANOP
.*
.* FIND_PARM OPERAND TERMINATED WITH SPACE
.* SET FIND_PARM_ERR IF ERROR
.*
.*         AENTRY FIND_PARM
             AGO   .APM_18_SKIP
.APM_18_FIND_PARM ANOP
&PARM    SETC ''
&FIND_PARM_ERR SETB 0
&IS_PARM SETA &IS_OP_END
.*         AWHILE (&IS_PARM LE K'&REC)
.AWH_12_T  ANOP
             AIF    (NOT(&IS_PARM LE K'&REC)).AWH_12_E
.*               AIF  ('&REC'(&IS_PARM,1) NE ' ')
                   AIF  (NOT('&REC'(&IS_PARM,1) NE ' ')).AIF_69_1
&I                    SETA ('&REC'(&IS_PARM,*) INDEX ' ')
.*                    AIF (&I GT 0 AND &IS_PARM+&I  LE K'&REC)
                        AIF (NOT(&I GT 0 AND &IS_PARM+&I  LE
K'&REC)).AIFX
                   _70_1
&PARM                     SETC '&REC'(&IS_PARM,&I-1)
.*                    AELSE
                        AGO   .AIF_70_E
.AIF_70_1               ANOP
&PARM                     SETC '&REC'(&IS_PARM,*)
.*                    AEND
.AIF_70_E               ANOP
.*                    AEXIT  AENTRY EXIT WITH PARM SET
                        AGO   .APM_18_E
.*               AEND
.AIF_69_1          ANOP
&IS_PARM         SETA &IS_PARM+1
.*         AEND
             AGO   .AWH_12_T
.AWH_12_E  ANOP
&FIND_PARM_ERR SETB 1
.*         AEND
```

```
.APM_18_E  ANOP
           AGO
(&APM_18_FIND_PARM).APM_18_1,.APM_18_2,.APM_18_3,.APM_X
               18_4
.APM_18_SKIP ANOP
.*
.* PUNCH LABEL WITH ANOP ALIGNED WITH AOP IF POSSIBLE
.*
.*        AENTRY PUNCH_LAB
            AGO   .APM_22_SKIP
.APM_22_PUNCH_LAB ANOP
&SPACES  SETA  &IS_OP+1-K'&PCH_REC
.*        AIF   (&SPACES LE 0)
            AIF   (NOT(&SPACES LE 0)).AIF_71_1
&SPACES         SETA  1
.*        AEND
.AIF_71_1  ANOP
&PCH_REC SETC '&PCH_REC'.(&SPACES)' '.'ANOP'
.*        APM  PUNCH_REC
&APM_19_PUNCH_REC SETA    34
            AGO   .APM_19_PUNCH_REC
.APM_19_34 ANOP
.*        AEND
            AGO
(&APM_22_PUNCH_LAB).APM_22_1,.APM_22_2,.APM_22_3,.APM_X

22_4,.APM_22_5,.APM_22_6,.APM_22_7,.APM_22_8,.APM_22_9,.X

APM_22_10,.APM_22_11,.APM_22_12,.APM_22_13,.APM_22_14,.AX
               PM_22_15,.APM_22_16,.APM_22_17
.APM_22_SKIP ANOP
.
.* PUNCH &PCH_REC WITH CONTINUATION FORMATTING AND RETURN TO CALLER
.* BASED ON &PUNCH_REC
.*
.*        AENTRY PUNCH_REC
            AGO   .APM_19_SKIP
.APM_19_PUNCH_REC ANOP
.*        AIF   (K'&PCH_REC GE 72)
            AIF   (NOT(K'&PCH_REC GE 72)).AIF_72_1
&TEXT           SETC (DOUBLE '&PCH_REC'(1,71))
                PUNCH '&TEXT.X',DDNAME=SYSUT2
&I              SETA  72
.*              AWHILE (K'&PCH_REC-&I GT 55)
.AWH_13_T         ANOP
```

```
                AIF    (NOT(K'&PCH_REC-&I GT 55)).AWH_13_E
&TEXT              SETC (DOUBLE '&PCH_REC'(&I,56))
                   PUNCH '               &TEXT.X',DDNAME=SYSUT2
&I                 SETA  &I+56
.*              AEND
                  AGO   .AWH_13_T
.AWH_13_E         ANOP
.*              AIF  (&I LE K'&PCH_REC)
                  AIF  (NOT(&I LE K'&PCH_REC)).AIF_73_1
&TEXT              SETC (DOUBLE '&PCH_REC'(&I,*))
                   PUNCH '               &TEXT',DDNAME=SYSUT2
.*              AEND
.AIF_73_1         ANOP
.*         AELSE
             AGO   .AIF_72_E
.AIF_72_1  ANOP
&TEXT           SETC (DOUBLE '&PCH_REC')
                PUNCH '&TEXT',DDNAME=SYSUT2
.*         AEND
.AIF_72_E  ANOP
.*         AEND
             AGO
(&APM_19_PUNCH_REC).APM_19_1,.APM_19_2,.APM_19_3,.APM_X

19_4,.APM_19_5,.APM_19_6,.APM_19_7,.APM_19_8,.APM_19_9,.X

APM_19_10,.APM_19_11,.APM_19_12,.APM_19_13,.APM_19_14,.AX

PM_19_15,.APM_19_16,.APM_19_17,.APM_19_18,.APM_19_19,.APX

M_19_20,.APM_19_21,.APM_19_22,.APM_19_23,.APM_19_24,.APMX

_19_25,.APM_19_26,.APM_19_27,.APM_19_28,.APM_19_29,.APM_X
                19_30,.APM_19_31,.APM_19_32,.APM_19_33,.APM_19_34
.APM_19_SKIP ANOP
.*
.* GEN_AIF - GENERATE AIF BRANCH
.*              1.  SET GEN_AIF_ERR TRUE/FALSE
.*              2.  BRANCH TRUE OR FALSE BASED ON GEN_AIF_TRUE
.*              3.  LABEL .&LVL_TYPE(&LVL)_&LVL_TCNT(&LVL)_&GEN_AIF_TAG
.*              4.  EXIT VIA COMPUTED AGO USING &GEN_AIF
.*
.*         AENTRY GEN_AIF
             AGO   .APM_23_SKIP
.APM_23_GEN_AIF ANOP
```

```
&GEN_AIF_ERR SETB 0
.*       APM   FIND_EXP
&APM_30_FIND_EXP SETA    2
           AGO   .APM_30_FIND_EXP
.APM_30_2  ANOP
.*       AIF   (&FIND_EXP_ERR)
           AIF   (NOT(&FIND_EXP_ERR)).AIF_74_1
&GEN_AIF_ERR   SETB 1
.*             AEXIT AENTRY
                 AGO   .APM_23_E
.*       AEND
.AIF_74_1  ANOP
&OP      SETC  (&IS_OP+1)' '.'AIF'.(&IS_EXP-&IS_OP-3)' '
&EXP     SETC  '&REC'(&IS_EXP,&IS_EXP_END-&IS_EXP+1)
&LAB     SETC
'.'.'&LVL_TYPE(&LVL)'(1,3).'_&LVL_TCNT(&LVL)_&GEN_AIF_TAX
               G'
.*       AIF   (NOT &GEN_AIF_TRUE)
           AIF   (NOT(NOT &GEN_AIF_TRUE)).AIF_75_1
&PCH_REC       SETC  '&OP.(NOT&EXP)&LAB'
.*       AELSE
           AGO   .AIF_75_E
.AIF_75_1  ANOP
&PCH_REC       SETC  '&OP&EXP&LAB'
.*       AEND
.AIF_75_E  ANOP
.CHK_AIF_COM ANOP
.*       AIF   (&IS_EXP_END LT K'&REC)
           AIF   (NOT(&IS_EXP_END LT K'&REC)).AIF_76_1
&PCH_REC       SETC  '&PCH_REC '.'&REC'(&IS_EXP_END+1,*) COMS
.*       AEND
.AIF_76_1  ANOP
.*       AEND
.APM_23_E  ANOP
           AGO
(&APM_23_GEN_AIF).APM_23_1,.APM_23_2,.APM_23_3,.APM_23X
                _4
.APM_23_SKIP ANOP
.*
.* FIND EXP - FIND EXPRESSION (..) AND SET IS_EXP AND IS_EXP_END
.*            SET FIND_EXP_ERR IF NOT FOUND
.*
.*       AENTRY FIND_EXP
           AGO   .APM_30_SKIP
.APM_30_FIND_EXP ANOP
```

```
&FIND_EXP_ERR SETB 0
&IS_EXP  SETA  ('&REC' INDEX '(')
.*       AIF   (&IS_EXP LE 0)
           AIF   (NOT(&IS_EXP LE 0)).AIF_77_1
&FIND_EXP_ERR  SETB 1
.*             AEXIT AENTRY
                 AGO   .APM_30_E
.*       AEND
.AIF_77_1  ANOP
&IS_EXP_END SETA &IS_EXP
&I       SETA  ('&REC'(&IS_EXP_END+1,*) INDEX ')')
.*       AWHILE (&I GT 0)
.AWH_14_T  ANOP
           AIF    (NOT(&I GT 0)).AWH_14_E
&IS_EXP_END    SETA &IS_EXP_END+&I
.*             AIF (&IS_EXP_END LT K'&REC)
                 AIF (NOT(&IS_EXP_END LT K'&REC)).AIF_78_1
&I                 SETA ('&REC'(&IS_EXP_END+1,*) INDEX ')')
.*             AELSE
                 AGO   .AIF_78_E
.AIF_78_1        ANOP
&I                 SETA 0
.*             AEND
.AIF_78_E        ANOP
.*       AEND
           AGO   .AWH_14_T
.AWH_14_E  ANOP
.*       AIF   (&IS_EXP_END EQ &IS_EXP)
           AIF   (NOT(&IS_EXP_END EQ &IS_EXP)).AIF_79_1
&FIND_EXP_ERR  SETB 1
.*       AEND
.AIF_79_1  ANOP
.*       AEND
.APM_30_E  ANOP
           AGO   (&APM_30_FIND_EXP).APM_30_1,.APM_30_2
.APM_30_SKIP ANOP
.*
.* ERR_MSG ISSUE ERROR MESSAGE AND COUNT ERRORS
.*
.*       AENTRY ERR_MSG
           AGO   .APM_3_SKIP
.APM_3_ERR_MSG ANOP
&ERRORS  SETA &ERRORS+1
         MNOTE  8,'ZSTRMAC ERROR &MSG AT LINE &LINE'
         PUNCH  ' MNOTE 8''ZSTRMAC ERROR &MSG',DDNAME=SYSUT2
```

```
.*       AEND
           AGO
(&APM_3_ERR_MSG).APM_3_1,.APM_3_2,.APM_3_3,.APM_3_4,.AX

PM_3_5,.APM_3_6,.APM_3_7,.APM_3_8,.APM_3_9,.APM_3_10,.APX

M_3_11,.APM_3_12,.APM_3_13,.APM_3_14,.APM_3_15,.APM_3_16X

,.APM_3_17,.APM_3_18,.APM_3_19,.APM_3_20,.APM_3_21,.APM_X

3_22,.APM_3_23,.APM_3_24,.APM_3_25,.APM_3_26,.APM_3_27,.X
               APM_3_28,.APM_3_29,.APM_3_30,.APM_3_31
.APM_3_SKIP ANOP
         MEND
         ZSTRMAC
         END
```