

zstrmac.txt

```
*****
* 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
* 09/17/08 RPI 911 CHANGE ASELECT TO ACASE AND APM TO ACALL AND
* SUPPORT LOWER CASE
*****
* 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. ACALL NAME > &ACALL_N SETA B
* > AGO .ACL_N
* > .ACL_N_B ANOP
* > .....
* 6. AENTRY NAME > .ACL_N ANOP
* > .....
* 7. AEXIT > AGO .ACL_N_E (EXIT NON AIF STRUCURE)
* > .....
* AEND > .ACL_N_E AGO (&ACALL_N).ACL_N_1,.ACL_N_2,
* > .ACL_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
```

zstrmac.txt

```
*          > .AUN_N_E ANOP
*          > .....
* 10. ACASE (EXP) >   AGO .ACS_N_AGO
* 11. AWHEN V1,V2 > .ACS_N_B1 ANOP  VN=(N,C'?',X'??', OR (V1,V2)
*          > .....
*   AWHEN V2     >   AGO .ACS_N_E
*          > .ACS_N_B2 ANOP
*          > .....
*   AELSE       >   AGO .ACS_N_E
*          > .ACS_N_X ANOP
*          > .....
*   AEND        >   AGO .ACS_N_E
*          > .ACS_N_G AGO (EXP).ACS_N_B1,.ACS_N_X,.ACS_N_B2
*          >   AGO .ACS_N_X
*          > .ACS_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,&ACALL_TOT
LCLA  &ACASE_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 ACALL/AENTRY NAME
LCLB  &FIND_PARM_ERR  SYNTAX ERROR FINDING FIRST PARM
LCLB  &FIND_EXP_ERR   SYNTAX ERROR FINDING (..) FOR AIF/ACASE
LCLB  &GET_VALUE_ERR  ERROR PARSING DEC, '?', OR X'??'
LCLA  &LVL            CURRENT LEVEL OF STRUCTURE
LCLC  &LVL_TYPE(50)  TYPE AIF/ACASE/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
```

zstrmac.txt

```

LCLC  &LVL_ACASE(50) ACASE COMPUTED AGO STATEMENT
LCLA  &LVL_ACASE_FIRST(50) ACASE FIRST WHEN VALUE 0-255
LCLA  &LVL_ACASE_LAST(50)  ACASE LAST  WHEN VALUE 0-255
LCLB  &LVL_AELSE(50)      AELSE BLOCK DEFINED FOR ACASE
LCLA  &IS_OP                START OF OPCODE
LCLA  &IS_OP_END            ENDOF OF OPCODE+1
LCLA  &IS_EXP               START OF AIF EXP (...)
LCLA  &ACALL_INDEX          INDEX TO ACALL/AENTRY VIA FIND_NAME
LCLA  &ACALL_NAME_TOT       TOTAL PERFORMED ROUTINES
LCLC  &ACALL_NAME(100)     NAMES OF PERFORMED ROUTINES
LCLA  &ACALL_CNT(100)      EXIT COUNT FOR ROUTINES
LCLB  &ACALL_DEF(100)      FLAG FOR DUP AND MISSING ERRORS

.*
.* READ SYUT1 AND OUTPUT SYSUT2 WITH STRUCTURED MACRO CODE
.*
.*      ACALL READ_REC
&ACALL_1_READ_REC SETA      1
                AGO      .ACL_1_READ_REC
.ACCL_1_1      ANOP
.*      AWHILE (NOT &EOF)
.ACWH_1_T      ANOP
                AIF      (NOT(NOT &EOF)).ACWH_1_E
.*      ACALL PROC_REC
&ACALL_2_PROC_REC SETA      1
                AGO      .ACL_2_PROC_REC
.ACCL_2_1      ANOP
.*      ACALL READ_REC
&ACALL_1_READ_REC SETA      2
                AGO      .ACL_1_READ_REC
.ACCL_1_2      ANOP
.*      AEND
                AGO      .ACWH_1_T
.ACWH_1_E      ANOP
&ACALL_INDEX SETA 1
.*      AWHILE (&ACALL_INDEX LE &ACALL_NAME_TOT)
.ACWH_2_T      ANOP
                AIF      (NOT(&ACALL_INDEX LE &ACALL_NAME_TOT)).ACWH_2_E
.*      AIF (NOT &ACALL_DEF(&ACALL_INDEX))
                AIF (NOT(NOT &ACALL_DEF(&ACALL_INDEX))).ACIF_1_1
&MSG          SETC 'MISSING AENTRY FOR &ACALL_NAME(&ACALL_INDEX)'
.*      ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      1
                AGO      .ACL_3_ERR_MSG
.ACCL_3_1      ANOP
.*      AEND

```

zstrmac.txt

```

.AIF_1_1          ANOP
&ACALL_INDEX     SETA &ACALL_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 ACALL      = &ACALL_TOT'
                MNOTE 'ZSTRMAC TOTAL ACASE     = &ACASE_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      .ACL_1_SKIP
.ACL_1_READ_REC ANOP
.*              ACALL READ_TEXT
&ACALL_4_READ_TEXT SETA      1
                AGO      .ACL_4_READ_TEXT
.ACL_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
.*              ACALL READ_TEXT
&ACALL_4_READ_TEXT SETA      2
                AGO      .ACL_4_READ_TEXT
.ACL_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 ' ')

```

```

                                zstrmac.txt
.AWH_3_T                        ANOP
                                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)
.*                               ACALL READ_TEXT
&ACALL_4_READ_TEXT              SETA      3
                                AGO      .ACL_4_READ_TEXT
.ACL_4_3                        ANOP
.*                               AEND
                                AGO      .AWH_3_T
.AWH_3_E                        ANOP
.*                               AIF      (NOT &EOF)
.*                               AIF      (NOT(NOT &EOF)).AIF_5_1
X                                AIF      (K'&TEXT GE 16
                                AND '&TEXT'(1,15) EQ (15)' ' )
X                                AIF      (NOT(K'&TEXT GE 16
                                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'
.*                               ACALL ERR_MSG
&ACALL_3_ERR_MSG              SETA      2
                                AGO      .ACL_3_ERR_MSG
.ACL_3_2                        ANOP
.*                               AEND
.AIF_6_E                        ANOP
.*                               AELSE
                                AGO      .AIF_5_E
.AIF_5_1                        ANOP
&MSG                            SETC 'END OF FILE ON CONTINUE'
.*                               ACALL ERR_MSG
&ACALL_3_ERR_MSG              SETA      3
                                AGO      .ACL_3_ERR_MSG

```

zstrmac.txt

```
.ACL_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   (&ACALL_1_READ_REC).ACL_1_1,.ACL_1_2
.ACL_1_SKIP ANOP
.*
.* READ LOGICAL LINE INTO &TEXT AND SET &EOF IF END OF FILE
.*
.*              AENTRY READ_TEXT
.*              AGO   .ACL_4_SKIP
.ACL_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   (&ACALL_4_READ_TEXT).ACL_4_1,.ACL_4_2,.ACL_4_3
.ACL_4_SKIP ANOP
.*
.* PROCESS REC BY SCANNING FOR A??? OPCODES AND GENERATING
.* COMMENT AND GENERATED CODE ELSE COPY REC
.*
.*              AENTRY PROC_REC
.*              AGO   .ACL_2_SKIP
.ACL_2_PROC_REC ANOP
.*              ACALL FIND_OPCODE
&ACALL_5_FIND_OPCODE SETA 1
.*              AGO   .ACL_5_FIND_OPCODE
```

zstrmac.txt

```

.ACL_5_1  ANOP
.*      AIF    ('&OPCODE'(1,1) NE 'A')
        AIF    (NOT('&OPCODE'(1,1) NE 'A')).AIF_8_1
.*
        ACALL  COPY_REC
&ACALL_6_COPY_REC SETA    1
        AGO    .ACL_6_COPY_REC
.ACL_6_1  ANOP
.*      AELSEIF ('&OPCODE' EQ 'AIF')
        AGO    .AIF_8_E
.AIF_8_1  ANOP
        AIF    (NOT('&OPCODE' EQ 'AIF')).AIF_8_2
.*
        ACALL  PROC_AIF
&ACALL_7_PROC_AIF SETA    1
        AGO    .ACL_7_PROC_AIF
.ACL_7_1  ANOP
.*      AELSEIF ('&OPCODE' EQ 'AELSE')
        AGO    .AIF_8_E
.AIF_8_2  ANOP
        AIF    (NOT('&OPCODE' EQ 'AELSE')).AIF_8_3
.*
        ACALL  PROC_AELSE
&ACALL_8_PROC_AELSE SETA    1
        AGO    .ACL_8_PROC_AELSE
.ACL_8_1  ANOP
.*      AELSEIF ('&OPCODE' EQ 'AELSEIF')
        AGO    .AIF_8_E
.AIF_8_3  ANOP
        AIF    (NOT('&OPCODE' EQ 'AELSEIF')).AIF_8_4
.*
        ACALL  PROC_AELSEIF
&ACALL_9_PROC_AELSEIF SETA    1
        AGO    .ACL_9_PROC_AELSEIF
.ACL_9_1  ANOP
.*      AELSEIF ('&OPCODE' EQ 'AEND')
        AGO    .AIF_8_E
.AIF_8_4  ANOP
        AIF    (NOT('&OPCODE' EQ 'AEND')).AIF_8_5
.*
        ACALL  PROC_AEND
&ACALL_10_PROC_AEND SETA    1
        AGO    .ACL_10_PROC_AEND
.ACL_10_1 ANOP
.*      AELSEIF ('&OPCODE' EQ 'ACALL')
        AGO    .AIF_8_E
.AIF_8_5  ANOP
        AIF    (NOT('&OPCODE' EQ 'ACALL')).AIF_8_6
.*
        ACALL  PROC_ACALL
&ACALL_11_PROC_ACALL SETA    1

```

zstrmac.txt

```
                AGO    .ACL_11_PROC_ACALL
.ACL_11_1      ANOP
.*            AELSEIF ('&OPCODE' EQ 'AENTRY')
                AGO    .AIF_8_E
.AIF_8_6      ANOP
                AIF      (NOT('&OPCODE' EQ 'AENTRY')).AIF_8_7
.*            ACALL PROC_AENTRY
&ACALL_12_PROC_AENTRY SETA    1
                AGO    .ACL_12_PROC_AENTRY
.ACL_12_1      ANOP
.*            AELSEIF ('&OPCODE' EQ 'AEXIT')
                AGO    .AIF_8_E
.AIF_8_7      ANOP
                AIF      (NOT('&OPCODE' EQ 'AEXIT')).AIF_8_8
.*            ACALL PROC_AEXIT
&ACALL_13_PROC_AEXIT SETA    1
                AGO    .ACL_13_PROC_AEXIT
.ACL_13_1      ANOP
.*            AELSEIF ('&OPCODE' EQ 'AWHILE')
                AGO    .AIF_8_E
.AIF_8_8      ANOP
                AIF      (NOT('&OPCODE' EQ 'AWHILE')).AIF_8_9
.*            ACALL PROC_AWHILE
&ACALL_14_PROC_AWHILE SETA    1
                AGO    .ACL_14_PROC_AWHILE
.ACL_14_1      ANOP
.*            AELSEIF ('&OPCODE' EQ 'AUNTIL')
                AGO    .AIF_8_E
.AIF_8_9      ANOP
                AIF      (NOT('&OPCODE' EQ 'AUNTIL')).AIF_8_10
.*            ACALL PROC_AUNTIL
&ACALL_15_PROC_AUNTIL SETA    1
                AGO    .ACL_15_PROC_AUNTIL
.ACL_15_1      ANOP
.*            AELSEIF ('&OPCODE' EQ 'ACASE')
                AGO    .AIF_8_E
.AIF_8_10     ANOP
                AIF      (NOT('&OPCODE' EQ 'ACASE')).AIF_8_11
.*            ACALL PROC_ACASE
&ACALL_16_PROC_ACASE SETA    1
                AGO    .ACL_16_PROC_ACASE
.ACL_16_1      ANOP
.*            AELSEIF ('&OPCODE' EQ 'AWHEN')
                AGO    .AIF_8_E
.AIF_8_11     ANOP
```

zstrmac.txt

```

                AIF      (NOT('&OPCODE' EQ 'AWHEN')).AIF_8_12
.*              ACALL  PROC_AWHEN
&ACALL_17_PROC_AWHEN SETA      1
                AGO      .ACL_17_PROC_AWHEN
.ACL_17_1       ANOP
.*              AELSE
                AGO      .AIF_8_E
.AIF_8_12       ANOP
.*              ACALL  COPY_REC
&ACALL_6_COPY_REC SETA      2
                AGO      .ACL_6_COPY_REC
.ACL_6_2        ANOP
.*              AEND
.AIF_8_E        ANOP
.*              AEND
                AGO      (&ACALL_2_PROC_REC).ACL_2_1
.ACL_2_SKIP     ANOP
.*
.* FIND_OPCODE - SET &OPCODE, &IS_OP, AND &IS_OP_END
.*
.*              AENTRY FIND_OPCODE
                AGO      .ACL_5_SKIP
.ACL_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      .ACL_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      .ACL_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

```

```

                                zstrmac.txt
                                AGO   .ACL_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   .ACL_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
.ACL_5_E                        ANOP
                                AGO   (&ACALL_5_FIND_OPCODE).ACL_5_1
.ACL_5_SKIP ANOP
.*
.*   COPY UNKNOWN RECORDS WITH :LABEL MOVED TO LABEL FIELD
.*
.*   AENTRY COPY_REC
                                AGO   .ACL_6_SKIP
.ACL_6_COPY_REC ANOP
.*   AIF  (K'&OPCODE GT 1
X
                                AND  &IS_OP_END LT K'&REC)
                                AIF  (NOT(K'&OPCODE GT 1
X

```

```

                                zstrmac.txt
                                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
.*                                ACALL FIND_PARM
&ACALL_18_FIND_PARM SETA      1
                                AGO  .ACL_18_FIND_PARM
.ACCL_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
.ACIF_16_1                      ANOP
&REC                              SETC '&REC'(&IS_OP+1,K'&OPCODE-1).(&SPACES)'
'X
                                .'&REC'(&IS_PARM,*)
.*                                AEND
.ACIF_15_1                      ANOP
.*                                AEND
.ACIF_14_1                      ANOP
.*                                AEND
.ACIF_13_1 ANOP
&PCH_REC SETC '&REC'
.*                                ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      1
                                AGO  .ACL_19_PUNCH_REC
.ACCL_19_1 ANOP
.*                                AEND
                                AGO  (&ACALL_6_COPY_REC).ACL_6_1,.ACL_6_2
.ACCL_6_SKIP ANOP
.*
.* AELSE - GEN MACRO COMMENT AND GEN AGO TO AEND AND LABEL FOR ALT.
BLK
.*
.*                                AENTRY PROC_AELSE
                                AGO  .ACL_8_SKIP
.ACCL_8_PROC_AELSE ANOP
&AELSE_TOT SETA &AELSE_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*                                ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      2
                                AGO  .ACL_19_PUNCH_REC
.ACCL_19_2 ANOP

```

zstrmac.txt

```

.*      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
.*      ACALL    PROC_AELSE_AIF
&ACALL_20_PROC_AELSE_AIF SETA      1
          AGO      .ACL_20_PROC_AELSE_AIF
.ACL_20_1  ANOP
.*      AELSEIF  (&LVL_TYPE(&LVL) EQ 'ACASE')
          AGO      .AIF_18_E
.AIF_18_1  ANOP
          AIF      (NOT(&LVL_TYPE(&LVL) EQ 'ACASE')).AIF_18_2
.*      ACALL    PROC_AELSE_ACASE
&ACALL_21_PROC_AELSE_ACASE SETA    1
          AGO      .ACL_21_PROC_AELSE_ACASE
.ACL_21_1  ANOP
.*      AELSE
          AGO      .AIF_18_E
.AIF_18_2  ANOP
&MSG      SETC    'INVALID AELSE TYPE &LVL_TYPE(&LVL)'
.*      ACALL    ERR_MSG
&ACALL_3_ERR_MSG SETA      4
          AGO      .ACL_3_ERR_MSG
.ACL_3_4   ANOP
.*      AEND
.AIF_18_E  ANOP
.*      AELSE
          AGO      .AIF_17_E
.AIF_17_1  ANOP
&MSG      SETC    'MISSING AIF OR ACASE'
.*      ACALL    ERR_MSG
&ACALL_3_ERR_MSG SETA      5
          AGO      .ACL_3_ERR_MSG
.ACL_3_5   ANOP
.*      AEND
.AIF_17_E  ANOP
.*      AEND
          AGO      (&ACALL_8_PROC_AELSE).ACL_8_1
.ACL_8_SKIP ANOP
.*
.* AELSE_AIF
.*
.*      AENTRY  PROC_AELSE_AIF
          AGO      .ACL_20_SKIP
.ACL_20_PROC_AELSE_AIF ANOP

```

zstrmac.txt

```
&LVL_TEND(&LVL) SETB 1 REQUEST AEND TO GEN END TARGET
&PCH_REC SETC (&IS_OP+1)' '.AGO .AIF_&LVL_TCNT(&LVL)_E'
.* ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA 3
AGO .ACL_19_PUNCH_REC
.ACL_19_3 ANOP
&PCH_REC SETC '.AIF_&LVL_TCNT(&LVL)_&LVL_BCNT(&LVL) '
.* ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA 1
AGO .ACL_22_PUNCH_LAB
.ACL_22_1 ANOP
&LVL_BCNT(&LVL) SETA 0 RESET TO INDICATE NO BLK LABEL REQ
.* AEND
AGO (&ACALL_20_PROC_AELSE_AIF).ACL_20_1
.ACL_20_SKIP ANOP
.*
.* AELSE_ACASE
.*
.* AENTRY PROC_AELSE_ACASE
AGO .ACL_21_SKIP
.ACL_21_PROC_AELSE_ACASE ANOP
.* AIF (&LVL_BCNT(&LVL) GT 0)
AIF (NOT(&LVL_BCNT(&LVL) GT 0)).AIF_19_1
&PCH_REC SETC (&IS_OP+1)' '.AGO .ACS_&LVL_TCNT(&LVL)_E'
.* ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA 4
AGO .ACL_19_PUNCH_REC
.ACL_19_4 ANOP
.* AEND
.AIF_19_1 ANOP
&LVL_AELSE(&LVL) SETB 1 INDICATE AELSE BLOCK DEFINED
&PCH_REC SETC '.ACS_&LVL_TCNT(&LVL)_X'
.* ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA 2
AGO .ACL_22_PUNCH_LAB
.ACL_22_2 ANOP
.* AEND
AGO (&ACALL_21_PROC_AELSE_ACASE).ACL_21_1
.ACL_21_SKIP ANOP
.*
.* AELSEIF - GEN MACRO COMMENT AND GEN AIF TO END OF BLK,CUR BLK LAB
.*
.* AENTRY PROC_AELSEIF
AGO .ACL_9_SKIP
.ACL_9_PROC_AELSEIF ANOP
```

zstrmac.txt

```
&AELSEIF_TOT SETA &AELSEIF_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      5
          AGO      .ACL_19_PUNCH_REC
.ACL_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'
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      6
          AGO      .ACL_19_PUNCH_REC
.ACL_19_6 ANOP
&PCH_REC      SETC '.AIF_&LVL_TCNT(&LVL)_&LVL_BCNT(&LVL) '
.*      ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA      3
          AGO      .ACL_22_PUNCH_LAB
.ACL_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) '
.*      ACALL GEN_AIF
&ACALL_23_GEN_AIF SETA      1
          AGO      .ACL_23_GEN_AIF
.ACL_23_1 ANOP
.*      AIF      (&GEN_AIF_ERR)
          AIF      (NOT(&GEN_AIF_ERR)).AIF_22_1
&MSG      SETC 'AELSEIF AIF ERROR'
.*      ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      6
          AGO      .ACL_3_ERR_MSG
.ACL_3_6 ANOP
.*      AELSE
          AGO      .AIF_22_E
.AIF_22_1 ANOP
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      7
          AGO      .ACL_19_PUNCH_REC
.ACL_19_7 ANOP
.*      AEND
.AIF_22_E ANOP
.*      AELSE
```

zstrmac.txt

```
                AGO    .AIF_21_E
.AIF_21_1      ANOP
&MSG          SETC  'AELSEIF MISSING AIF ERROR'
.*           ACALL  ERR_MSG
&ACALL_3_ERR_MSG SETA    7
                AGO    .ACL_3_ERR_MSG
.ACL_3_7      ANOP
.*           AEND
.AIF_21_E     ANOP
.*           AELSE
                AGO    .AIF_20_E
.AIF_20_1     ANOP
&MSG          SETC  'AELSEIF MISSING AIF ERROR'
.*           ACALL  ERR_MSG
&ACALL_3_ERR_MSG SETA    8
                AGO    .ACL_3_ERR_MSG
.ACL_3_8      ANOP
.*           AEND
.AIF_20_E     ANOP
.*           AEND
                AGO    (&ACALL_9_PROC_AELSEIF).ACL_9_1
.ACL_9_SKIP   ANOP
.*
.* AEND - GEN TERMINATION FOR AENTRY,AIF,ACASE,AUNTIL,AWHILE
.*
.*           AENTRY PROC_AEND
                AGO    .ACL_10_SKIP
.ACL_10_PROC_AEND ANOP
&AEND_TOT SETA &AEND_TOT+1
&PCH_REC SETC '.*'. '&REC'(3,*)
.*           ACALL  PUNCH_REC
&ACALL_19_PUNCH_REC SETA    8
                AGO    .ACL_19_PUNCH_REC
.ACL_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
.*           ACALL  PROC_AEND_AIF
&ACALL_24_PROC_AEND_AIF SETA    1
                AGO    .ACL_24_PROC_AEND_AIF
.ACL_24_1     ANOP
.*           AELSEIF (&LVL_TYPE(&LVL) EQ 'AWHILE')
                AGO    .AIF_24_E
.AIF_24_1     ANOP
```

```

                                zstrmac.txt
                                AIF      (NOT(&LVL_TYPE(&LVL) EQ 'AWHILE')).AIF_24_2
.*                               ACALL  PROC_AEND_AWHILE
&ACALL_25_PROC_AEND_AWHILE SETA   1
                                AGO     .ACL_25_PROC_AEND_AWHILE
.ACCL_25_1                       ANOP
.*                               AELSEIF (&LVL_TYPE(&LVL) EQ 'ACASE')
                                AGO     .AIF_24_E
.ACIF_24_2                       ANOP
                                AIF      (NOT(&LVL_TYPE(&LVL) EQ 'ACASE')).AIF_24_3
.*                               ACALL  PROC_AEND_ACASE
&ACALL_26_PROC_AEND_ACASE SETA   1
                                AGO     .ACL_26_PROC_AEND_ACASE
.ACCL_26_1                       ANOP
.*                               AELSEIF (&LVL_TYPE(&LVL) EQ 'AENTRY')
                                AGO     .AIF_24_E
.ACIF_24_3                       ANOP
                                AIF      (NOT(&LVL_TYPE(&LVL) EQ 'AENTRY')).AIF_24_4
.*                               ACALL  PROC_AEND_AENTRY
&ACALL_27_PROC_AEND_AENTRY SETA   1
                                AGO     .ACL_27_PROC_AEND_AENTRY
.ACCL_27_1                       ANOP
.*                               AELSEIF (&LVL_TYPE(&LVL) EQ 'AUNTIL')
                                AGO     .AIF_24_E
.ACIF_24_4                       ANOP
                                AIF      (NOT(&LVL_TYPE(&LVL) EQ 'AUNTIL')).AIF_24_5
.*                               ACALL  PROC_AEND_AUNTIL
&ACALL_28_PROC_AEND_AUNTIL SETA   1
                                AGO     .ACL_28_PROC_AEND_AUNTIL
.ACCL_28_1                       ANOP
.*                               AELSE
                                AGO     .AIF_24_E
.ACIF_24_5                       ANOP
&MSG                               SETC 'AEND INVALID TYPE &LVL_TYPE(&LVL)'
.*                               ACALL  ERR_MSG
&ACALL_3_ERR_MSG                 SETA   9
                                AGO     .ACL_3_ERR_MSG
.ACCL_3_9                       ANOP
.*                               AEND
.ACIF_24_E                       ANOP
.*                               AELSE
                                AGO     .AIF_23_E
.ACIF_23_1                       ANOP
&MSG                               SETC 'AEND MISSING AIF OR OTHER STRUCTURE'
.*                               ACALL  ERR_MSG
&ACALL_3_ERR_MSG SETA   10

```

```

                                zstrmac.txt
                                AGO    .ACL_3_ERR_MSG
.ACCL_3_10                      ANOP
.*                               AEND
.ACIF_23_E                      ANOP
.*                               AEND
                                AGO    (&ACALL_10_PROC_AEND).ACL_10_1
.ACCL_10_SKIP                   ANOP
.*
.* AEND_AENTRY
.*
.*                               AENTRY PROC_AEND_AENTRY
                                AGO    .ACL_27_SKIP
.ACCL_27_PROC_AEND_AENTRY      ANOP
&ACALL_INDEX SETA &LVL_BCNT(&LVL)
.*                               AIF    (&ACALL_CNT(&ACALL_INDEX) GT 0)
                                AIF    (NOT(&ACALL_CNT(&ACALL_INDEX) GT 0)).AIF_25_1
.*                               AIF    (&LVL_TEND(&LVL))
                                AIF    (NOT(&LVL_TEND(&LVL))).AIF_26_1
&PCH_REC                        SETC  '.ACL &ACALL_INDEX._E'
.*                               ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB           SETA    4
                                AGO    .ACL_22_PUNCH_LAB
.ACCL_22_4                      ANOP
.*                               AEND
.ACIF_26_1                      ANOP
&PCH_REC                        SETC  (&IS_OP+1)' '.AGO
(&&ACALL_&ACALL_INDEX._&ACALL_X
                                NAME(&ACALL_INDEX)).ACL_&ACALL_INDEX._1'
&I                               SETA  2
.*                               AWHILE (&I LE &ACALL_CNT(&ACALL_INDEX))
.ACWH_4_T                       ANOP
                                AIF    (NOT(&I LE &ACALL_CNT(&ACALL_INDEX))).ACWH_4_E
&PCH_REC                        SETC  '&PCH_REC, .ACL &ACALL_INDEX._&I'
&I                               SETA  &I+1
.*                               AEND
                                AGO    .ACWH_4_T
.ACWH_4_E                       ANOP
.*                               ACALL PUNCH_REC
&ACALL_19_PUNCH_REC           SETA    9
                                AGO    .ACL_19_PUNCH_REC
.ACCL_19_9                      ANOP
.*                               AELSE
                                AGO    .AIF_25_E
.ACIF_25_1                     ANOP
&MSG                            SETC  'AENTRY &ACALL_NAME(&ACALL_INDEX) NOT USED'

```

zstrmac.txt

```
.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA    11
                AGO    .ACL_3_ERR_MSG
.ACL_3_11      ANOP
.*          AEND
.AIF_25_E     ANOP
&PCH_REC SETC '.ACL_&ACALL_INDEX._SKIP'
.*          ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA    5
                AGO    .ACL_22_PUNCH_LAB
.ACL_22_5     ANOP
&LVL        SETA  &LVL-1      CURRENT LEVEL
.*          AEND
                AGO    (&ACALL_27_PROC_AEND_AENTRY).ACL_27_1
.ACL_27_SKIP ANOP
.*
.* AEND_AIF
.*
.*          AENTRY PROC_AEND_AIF
                AGO    .ACL_24_SKIP
.ACL_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)'
.*          ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA    6
                AGO    .ACL_22_PUNCH_LAB
.ACL_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'
.*          ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA    7
                AGO    .ACL_22_PUNCH_LAB
.ACL_22_7     ANOP
.*          AEND
.AIF_28_1     ANOP
&LVL        SETA  &LVL-1      CURRENT LEVEL
.*          AEND
                AGO    (&ACALL_24_PROC_AEND_AIF).ACL_24_1
.ACL_24_SKIP ANOP
.*
.* AEND_AUNTIL
```

zstrmac.txt

```

.*
.*      AENTRY PROC_AEND_AUNTIL
        AGO      .ACL_28_SKIP
.ACCL_28_PROC_AEND_AUNTIL ANOP
&PCH_REC SETC (&IS_OP+1)' '.AGO      .AUN_&LVL_TCNT(&LVL)_T'
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      10
        AGO      .ACL_19_PUNCH_REC
.ACCL_19_10 ANOP
&PCH_REC SETC '.AUN_&LVL_TCNT(&LVL)_E'
.*      ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA      8
        AGO      .ACL_22_PUNCH_LAB
.ACCL_22_8 ANOP
&LVL      SETA  &LVL-1      CURRENT LEVEL
.*      AEND
        AGO      (&ACALL_28_PROC_AEND_AUNTIL).ACL_28_1
.ACCL_28_SKIP ANOP
.*
.* AEND_AWHILE
.*
.*      AENTRY PROC_AEND_AWHILE
        AGO      .ACL_25_SKIP
.ACCL_25_PROC_AEND_AWHILE ANOP
&PCH_REC SETC (&IS_OP+1)' '.AGO      .AWH_&LVL_TCNT(&LVL)_T'
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      11
        AGO      .ACL_19_PUNCH_REC
.ACCL_19_11 ANOP
&PCH_REC SETC '.AWH_&LVL_TCNT(&LVL)_E'
.*      ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA      9
        AGO      .ACL_22_PUNCH_LAB
.ACCL_22_9 ANOP
&LVL      SETA  &LVL-1      CURRENT LEVEL
.*      AEND
        AGO      (&ACALL_25_PROC_AEND_AWHILE).ACL_25_1
.ACCL_25_SKIP ANOP
.*
.* AEND_ACASE
.*
.*      AENTRY PROC_AEND_ACASE
        AGO      .ACL_26_SKIP
.ACCL_26_PROC_AEND_ACASE ANOP
.*      AIF      (&LVL_BCNT(&LVL) GT 0)

```

```

                                zstrmac.txt
AIF      (NOT(&LVL_BCNT(&LVL) GT 0)).AIF_29_1
&PCH_REC SETC (&IS_OP+1)' '.AGO      .ACS_&LVL_TCNT(&LVL)_E'
.*
ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      12
                                AGO      .ACL_19_PUNCH_REC
.ACCL_19_12 ANOP
&PCH_REC SETC '.ACS_&LVL_TCNT(&LVL)_G'
.*
ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA      10
                                AGO      .ACL_22_PUNCH_LAB
.ACCL_22_10 ANOP
.*
AIF      (&LVL_AELSE(&LVL))
                                AIF      (NOT(&LVL_AELSE(&LVL))).AIF_30_1
&ELSE_LAB SETC '.ACS_&LVL_TCNT(&LVL)_X'
.*
AELSE
                                AGO      .AIF_30_E
.AIF_30_1 ANOP
&ELSE_LAB SETC '.ACS_&LVL_TCNT(&LVL)_E'
.*
AEND
.AIF_30_E ANOP
&PCH_REC SETC '&LVL_ACASE(&LVL)'
.*
AIF      (&LVL_ACASE_FIRST(&LVL) NE 1)
                                AIF      (NOT(&LVL_ACASE_FIRST(&LVL) NE 1)).AIF_31_1
&OFFSET SETC '+1-&LVL_ACASE_FIRST(&LVL)'
&PCH_REC SETC '&PCH_REC'(1,K'&PCH_REC-1).'&OFFSET)'
.*
AEND
.AIF_31_1 ANOP
&VAL_BLK SETC 'ACASE_&LVL_TCNT(&LVL)_VAL_BLK'
&VALUE SETA &LVL_ACASE_FIRST(&LVL)
&COMMA SETC ''
.*
AWHILE (&VALUE LE &LVL_ACASE_LAST(&LVL))
.AWH_5_T ANOP
                                AIF      (NOT(&VALUE LE &LVL_ACASE_LAST(&LVL))).AWH_5_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..ACS_&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'

```

```

                                zstrmac.txt
&COMMA                          SETC  ', '
.*                               AEND
.AIF_32_E                        ANOP
&VALUE                          SETA  &VALUE+1
.*                               AEND
                                AGO   .AWH_5_T
.AWH_5_E                        ANOP
.*                               ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      13
                                AGO   .ACL_19_PUNCH_REC
.ACL_19_13                      ANOP
.*                               AIF   (&LVL_AELSE(&LVL))
                                AIF   (NOT(&LVL_AELSE(&LVL))).AIF_33_1
&PCH_REC                        SETC (&IS_OP+1)' '. 'AGO   .ACS_&LVL_TCNT(&LVL)_X'
.*                               ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      14
                                AGO   .ACL_19_PUNCH_REC
.ACL_19_14                      ANOP
.*                               AEND
.AIF_33_1                      ANOP
&PCH_REC                        SETC '.ACS_&LVL_TCNT(&LVL)_E'
.*                               ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA      11
                                AGO   .ACL_22_PUNCH_LAB
.ACL_22_11                      ANOP
&LVL                            SETA  &LVL-1      CURRENT LEVEL
.*                               AELSE
                                AGO   .AIF_29_E
.AIF_29_1                      ANOP
&MSG                            SETC 'NO WHEN FOUND FOR ACASE'
.*                               ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA         12
                                AGO   .ACL_3_ERR_MSG
.ACL_3_12                      ANOP
.*                               AEND
.AIF_29_E                      ANOP
.*                               AEND
                                AGO   (&ACALL_26_PROC_AEND_ACASE).ACL_26_1
.ACL_26_SKIP ANOP
.*
.* AENTRY - GEN AGO BRANCH AROUND PENTRY/PEND AND LABEL FOR ENTRY
.*
.*                               AENTRY PROC_AENTRY
                                AGO   .ACL_12_SKIP
.ACL_12_PROC_AENTRY ANOP

```

zstrmac.txt

```
&AENTRY_TOT SETA &AENTRY_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      15
          AGO      .ACL_19_PUNCH_REC
.ACL_19_15 ANOP
.*      ACALL FIND_NAME
&ACALL_29_FIND_NAME SETA      1
          AGO      .ACL_29_FIND_NAME
.ACL_29_1 ANOP
.*      AIF      (&FIND_NAME_ERR)
          AIF      (NOT(&FIND_NAME_ERR)).AIF_34_1
&MSG          SETC 'AENTRY NAME NOT FOUND'
.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      13
          AGO      .ACL_3_ERR_MSG
.ACL_3_13 ANOP
.*      AELSEIF (&ACALL_DEF(&ACALL_INDEX))
          AGO      .AIF_34_E
.AIF_34_1 ANOP
          AIF      (NOT(&ACALL_DEF(&ACALL_INDEX))).AIF_34_2
&MSG          SETC 'AENTRY DUPLICATE NAME FOUND - &NAME'
.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      14
          AGO      .ACL_3_ERR_MSG
.ACL_3_14 ANOP
.*      AELSE
          AGO      .AIF_34_E
.AIF_34_2 ANOP
&ACALL_DEF(&ACALL_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 &ACALL_INDEX      SAVE FOR AEND
&PCH_REC          SETC (&IS_OP+1)' '.'AGO      .ACL_&ACALL_INDEX._SKIP'
.*          ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      16
          AGO      .ACL_19_PUNCH_REC
.ACL_19_16 ANOP
&PCH_REC          SETC '.ACL_&ACALL_INDEX._&ACALL_NAME(&ACALL_INDEX)'
.*          ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA      12
          AGO      .ACL_22_PUNCH_LAB
.ACL_22_12 ANOP
```

zstrmac.txt

```

.*          AEND
.AIF_34_E  ANOP
.*          AEND
            AGO    (&ACALL_12_PROC_AENTRY).ACL_12_1
.ACL_12_SKIP ANOP
.*
.* AEXIT - EXIT TO FIRST MATCHING TYPE FOUND
.*
.*          AENTRY PROC_AEXIT
            AGO    .ACL_13_SKIP
.ACL_13_PROC_AEXIT ANOP
&AEXIT_TOT SETA &AEXIT_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*          ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA    17
            AGO    .ACL_19_PUNCH_REC
.ACL_19_17 ANOP
.*          ACALL FIND_PARM
&ACALL_18_FIND_PARM SETA    2
            AGO    .ACL_18_FIND_PARM
.ACL_18_2  ANOP
.*          AIF  (&FIND_PARM_ERR)
            AIF  (NOT(&FIND_PARM_ERR)).AIF_35_1
&MSG          SETC 'AEXIT TYPE PARM NOT FOUND'
.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA    15
            AGO    .ACL_3_ERR_MSG
.ACL_3_15    ANOP
.*          AEXIT AENTRY
            AGO    .ACL_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

```

zstrmac.txt

```

.*                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
&ACALL_INDEX     SETA &LVL_BCNT(&EXIT_LVL)
&PCH_REC         SETC (&IS_OP+1)' '.AGO   .ACL_&ACALL_INDEX._E'
.*              ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA    18
                AGO   .ACL_19_PUNCH_REC
.ACL_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'
.*              ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA    19
                AGO   .ACL_19_PUNCH_REC
.ACL_19_19       ANOP
.*              AEND
.AIF_38_E        ANOP
.*              AELSE
                AGO   .AIF_37_E
.AIF_37_1        ANOP
&MSG             SETC 'AEXIT NOT WITHIN AENTRY, AWHILE, ACASE'
.*              ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA    16
                AGO   .ACL_3_ERR_MSG
.ACL_3_16        ANOP
.*              AEND
.AIF_37_E        ANOP
.*              AEND
.ACL_13_E        ANOP
                AGO   (&ACALL_13_PROC_AEXIT).ACL_13_1
.ACL_13_SKIP     ANOP
.*
.* AIF - GEN MACRO COMMENT AND AIF TO GENERATED END LABEL AT NEXT
LEVEL

```

zstrmac.txt

```

.*
.*      AENTRY PROC_AIF
        AGO      .ACL_7_SKIP
.ACCL_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,*)
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      20
        AGO      .ACL_19_PUNCH_REC
.ACCL_19_20 ANOP
&GEN_AIF_TRUE SETB 0          GEN BRANCH IF FALSE
&GEN_AIF_TAG SETC '&LVL_BCNT(&LVL) '
.*      ACALL GEN_AIF
&ACALL_23_GEN_AIF SETA      2
        AGO      .ACL_23_GEN_AIF
.ACCL_23_2 ANOP
.*      AIF      (&GEN_AIF_ERR)
        AIF      (NOT(&GEN_AIF_ERR)).AIF_39_1
&MSG          SETC 'AIF EXPRESSION SYNTAX ERROR'
.*      ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      17
        AGO      .ACL_3_ERR_MSG
.ACCL_3_17 ANOP
.*      AELSE
        AGO      .AIF_39_E
.AIF_39_1 ANOP
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      21
        AGO      .ACL_19_PUNCH_REC
.ACCL_19_21 ANOP
.*      AEND
.AIF_39_E ANOP
.*      AEND
        AGO      (&ACALL_7_PROC_AIF).ACL_7_1
.ACCL_7_SKIP ANOP
.*
.* ACALL - GEN AGO TO PERFORMED ROUTINE
.*
.*      AENTRY PROC_ACALL
        AGO      .ACL_11_SKIP

```

zstrmac.txt

```

.ACL_11_PROC_ACALL ANOP
&ACALL_TOT SETA &ACALL_TOT+1
&PCH_REC SETC '.*'.'&REC'(3,*)
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      22
          AGO      .ACL_19_PUNCH_REC
.ACL_19_22 ANOP
.*      ACALL FIND_NAME
&ACALL_29_FIND_NAME SETA      2
          AGO      .ACL_29_FIND_NAME
.ACL_29_2 ANOP
.*      AIF      (&FIND_NAME_ERR)
          AIF      (NOT(&FIND_NAME_ERR)).AIF_40_1
&MSG          SETC 'ACALL NAME SYNTAX ERROR'
.*      ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      18
          AGO      .ACL_3_ERR_MSG
.ACL_3_18          ANOP
.*      AELSE
          AGO      .AIF_40_E
.AIF_40_1 ANOP
&ACALL_CNT(&ACALL_INDEX) SETA &ACALL_CNT(&ACALL_INDEX)+1
&PCH_REC          SETC '&&ACALL_&ACALL_INDEX._&ACALL_NAME(&ACALL_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
&ACALL_CNT(&ACALL_X
          INDEX)'
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      23
          AGO      .ACL_19_PUNCH_REC
.ACL_19_23          ANOP
&PCH_REC          SETC (&IS_OP+1)' '.'AGO
.ACL_&ACALL_INDEX._&ACALL_NAME
          (&ACALL_INDEX)'
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      24
          AGO      .ACL_19_PUNCH_REC
.ACL_19_24          ANOP
&PCH_REC          SETC '.ACL_&ACALL_INDEX._&ACALL_CNT(&ACALL_INDEX)'
.*      ACALL PUNCH_REC

```

```

                                zstrmac.txt
&ACALL_22_PUNCH_LAB SETA      13
                                AGO      .ACL_22_PUNCH_LAB
.ACCL_22_13                    ANOP
.*                               AEND
.ACIF_40_E                    ANOP
.*                               AEND
                                AGO      (&ACALL_11_PROC_ACALL).ACL_11_1
.ACCL_11_SKIP                  ANOP
.*
.* ACASE - GEN AGO TO .ACS_N_AGO AND SAVE AGO EXPRESSION
.*
.*                               AENTRY PROC_ACASE
                                AGO      .ACL_16_SKIP
.ACCL_16_PROC_ACASE           ANOP
&ACASE_TOT SETA  &ACASE_TOT+1    ACASE COUNTER
&LVL             SETA  &LVL+1     CURRENT LEVEL
&LVL_TYPE(&LVL) SETC 'ACASE' CURRENT LEVEL TYPE
&LVL_TCNT(&LVL) SETA &ACASE_TOT ACASE INSTANCE
&LVL_BCNT(&LVL) SETA 0 RESET ACASE AWHEN BLOCKS
&LVL_AELSE(&LVL) SETB 0 ASSUME NO AELSE BLOCK
&VAL_BLK SETC   'ACASE_&LVL_TCNT(&LVL)_VAL_BLK'
                                LCLA &(&VAL_BLK)(256)
&LVL_ACASE_FIRST(&LVL) SETA 257
&LVL_ACASE_LAST(&LVL) SETA -1
&PCH_REC SETC   '.*'. '&REC'(3,*)
.*                               ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      25
                                AGO      .ACL_19_PUNCH_REC
.ACCL_19_25                    ANOP
.*                               ACALL FIND_EXP
&ACALL_30_FIND_EXP SETA      1
                                AGO      .ACL_30_FIND_EXP
.ACCL_30_1                    ANOP
.*                               AIF      (&FIND_EXP_ERR)
                                AIF      (NOT(&FIND_EXP_ERR)).AIF_42_1
&MSG                            SETC 'ACASE EXPRESSION ERROR'
.*                               ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      19
                                AGO      .ACL_3_ERR_MSG
.ACCL_3_19                    ANOP
.*                               AELSE
                                AGO      .AIF_42_E
.ACIF_42_1                    ANOP
&LVL_ACASE(&LVL) SETC (&IS_OP+1)' '.'AGO
'.'&REC'(&IS_EXP,&IS_EXP_ENDX

```

zstrmac.txt

```

-&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      .ACS_&LVL_TCNT(&LVL)_G'
.*          ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA 26
            AGO      .ACL_19_PUNCH_REC
.ACL_19_26  ANOP
.*          AEND
.AIF_42_E   ANOP
.*          AEND
            AGO      (&ACALL_16_PROC_ACASE).ACL_16_1
.ACL_16_SKIP ANOP
.*
.* AUNTIL - GEN AGO TO BLOCK, THEN LABEL TEST AIF TO EXIT
.*
.*          AENTRY PROC_AUNTIL
            AGO      .ACL_15_SKIP
.ACL_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,*)
.*          ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA 27
            AGO      .ACL_19_PUNCH_REC
.ACL_19_27 ANOP
&PCH_REC SETC (&IS_OP+1)' '.AGO      .AUN_&LVL_TCNT(&LVL)'
.*          ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA 28
            AGO      .ACL_19_PUNCH_REC
.ACL_19_28 ANOP
&PCH_REC SETC '.AUN_&LVL_TCNT(&LVL)_T'
.*          ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA 14
            AGO      .ACL_22_PUNCH_LAB
.ACL_22_14 ANOP

```

zstrmac.txt

```
&GEN_AIF_TRUE SETB 1 GEN BRANCH IF TRUE
&GEN_AIF_TAG SETC 'E'
.* ACALL GEN_AIF
&ACALL_23_GEN_AIF SETA 3
AGO .ACL_23_GEN_AIF
.ACL_23_3 ANOP
.* AIF (&GEN_AIF_ERR)
AIF (NOT(&GEN_AIF_ERR)).AIF_43_1
&MSG SETC 'AUNTIL EXPRESSION ERROR'
.* ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA 20
AGO .ACL_3_ERR_MSG
.ACL_3_20 ANOP
.* AELSE
AGO .AIF_43_E
.AIF_43_1 ANOP
.* ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA 29
AGO .ACL_19_PUNCH_REC
.ACL_19_29 ANOP
.* AEND
.AIF_43_E ANOP
&PCH_REC SETC '.AUN_&LVL_TCNT(&LVL)'
.* ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA 15
AGO .ACL_22_PUNCH_LAB
.ACL_22_15 ANOP
.* AEND
AGO (&ACALL_15_PROC_AUNTIL).ACL_15_1
.ACL_15_SKIP ANOP
.*
.* AWHEN - GEN .ACS_N_I LABEL FOR INDEX AND UPDATE INDEX VAL_BLK
.*
.* AENTRY PROC_AWHEN
AGO .ACL_17_SKIP
.ACL_17_PROC_AWHEN ANOP
&PCH_REC SETC '.*'. '&REC'(3,*)
.* ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA 30
AGO .ACL_19_PUNCH_REC
.ACL_19_30 ANOP
&AWHEN_TOT SETA &AWHEN_TOT+1
.* AIF (&LVL GE 1)
AIF (NOT(&LVL GE 1)).AIF_44_1
&VAL_BLK SETC 'ACASE_&LVL_TCNT(&LVL)_VAL_BLK'
```

```

                                zstrmac.txt
.*          AIF (&LVL_TYPE(&LVL) EQ 'ACASE')
          AIF (NOT(&LVL_TYPE(&LVL) EQ 'ACASE')).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
.&ACS_&LVL_TCNT(&LVLX
          )_E'
.*          ACALL PUNCH_REC
&ACALL_19_PUNCH_REC          SETA 31
          AGO .ACL_19_PUNCH_REC
.&ACL_19_31          ANOP
.*          AEND
.&AIF_46_1          ANOP
&LVL_BCNT(&LVL)          SETA &LVL_BCNT(&LVL)+1
.*          ACALL FIND_PARM
&ACALL_18_FIND_PARM          SETA 3
          AGO .ACL_18_FIND_PARM
.&ACL_18_3          ANOP
.*          AIF (&FIND_PARM_ERR)
          AIF (NOT(&FIND_PARM_ERR)).AIF_47_1
&MSG          SETC 'AWHEN VALUE ERROR'
.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG          SETA 21
          AGO .ACL_3_ERR_MSG
.&ACL_3_21          ANOP
.*          AELSE
          AGO .AIF_47_E
.&AIF_47_1          ANOP
.*          ACALL PROC_AWHEN_VALUES
&ACALL_31_PROC_AWHEN_VALUES          SETA 1
          AGO .ACL_31_PROC_AWHEN_VALUES
.&ACL_31_1          ANOP
.*          AEND
.&AIF_47_E          ANOP
&PCH_REC          SETC '.ACS_&LVL_TCNT(&LVL)_&LVL_BCNT(&LVL)'
.*          ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB          SETA 16
          AGO .ACL_22_PUNCH_LAB
.&ACL_22_16          ANOP
.*          AELSE
          AGO .AIF_45_E
.&AIF_45_1          ANOP
&MSG          SETC 'AWHEN MISSING ACASE'

```

zstrmac.txt

```

.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG   SETA   22
                   AGO   .ACL_3_ERR_MSG
.ACCL_3_22         ANOP
.*          AEND
.ACIF_45_E         ANOP
.*          AELSE
                   AGO   .AIF_44_E
.ACIF_44_1        ANOP
&MSG              SETC 'AWHEN MISSING ACASE'
.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA   23
                   AGO   .ACL_3_ERR_MSG
.ACCL_3_23        ANOP
.*          AEND
.ACIF_44_E        ANOP
.*          AEND
                   AGO   (&ACALL_17_PROC_AWHEN).ACL_17_1
.ACCL_17_SKIP     ANOP
.*
.* PROC_WHEN_VALUES V1,V2,(V3,V4) WHERE VN = DEC, C'?', OR X'??'
.*
.*          AENTRY PROC_AWHEN_VALUES
                   AGO   .ACL_31_SKIP
.ACCL_31_PROC_AWHEN_VALUES ANOP
&VALUE_CNT SETA 0
.*          AWHILE (&IS_PARM LE K'&REC)
.ACWH_8_T         ANOP
                   AIF   (NOT(&IS_PARM LE K'&REC)).ACWH_8_E
.*          ACASE (C2A('&REC'(&IS_PARM,1)))
                   AGO   .ACS_1_G
.*          AWHEN C'(' SET RANGE (V1,V2)
.ACCL_1_1         ANOP
&IS_PARM          SETA &IS_PARM+1
.*          ACALL GET_VALUE
&ACALL_32_GET_VALUE SETA   1
                   AGO   .ACL_32_GET_VALUE
.ACCL_32_1        ANOP
.*          AIF   (&GET_VALUE_ERR)
                   AIF   (NOT(&GET_VALUE_ERR)).AIF_48_1
&MSG              SETC 'INVALID RANGE VALUE'
.*          ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA   24
                   AGO   .ACL_3_ERR_MSG
.ACCL_3_24        ANOP

```

```

                                zstrmac.txt
.*                                AEXIT AENTRY      EXIT AFTER VALUE ERROR
                                AGO      .ACL_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 ,'
.*                                ACALL ERR_MSG
&ACALL_3_ERR_MSG                  SETA      25
                                AGO      .ACL_3_ERR_MSG
.ACL_3_25                          ANOP
.*                                AEXIT AENTRY
                                AGO      .ACL_31_E

.*                                AEND
.AIF_49_1                        ANOP
&IS_PARM                          SETA &IS_PARM+1
.*                                ACALL GET_VALUE
&ACALL_32_GET_VALUE              SETA      2
                                AGO      .ACL_32_GET_VALUE
.ACL_32_2                          ANOP
.*                                AIF (&GET_VALUE_ERR)
                                AIF (NOT(&GET_VALUE_ERR)).AIF_50_1
&MSG                              SETC 'INVALID RANGE VALUE'
.*                                ACALL ERR_MSG
&ACALL_3_ERR_MSG                  SETA      26
                                AGO      .ACL_3_ERR_MSG
.ACL_3_26                          ANOP
.*                                AEXIT AENTRY      EXIT AFTER VALUE ERROR
                                AGO      .ACL_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 )'
.*                                ACALL ERR_MSG
&ACALL_3_ERR_MSG                  SETA      27
                                AGO      .ACL_3_ERR_MSG
.ACL_3_27                          ANOP
.*                                AEXIT AENTRY
                                AGO      .ACL_31_E

.*                                AEND

```

```

                                zstrmac.txt
.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
.*                               ACALL SET_VAL_BLK
&ACALL_33_SET_VAL_BLK          SETA      1
                                AGO      .ACL_33_SET_VAL_BLK
.ACL_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      .ACS_1_E
.ACS_1_2                      ANOP
.*                               AEXIT AWHILE
                                AGO      .AWH_8_E
.*                               AWHEN C', '
                                AGO      .ACS_1_E
.ACS_1_3                      ANOP
&IS_PARM                        SETA &IS_PARM+1
.*                               AELSE
                                AGO      .ACS_1_E
.ACS_1_X                      ANOP
.*                               ACALL GET_VALUE
&ACALL_32_GET_VALUE          SETA      3
                                AGO      .ACL_32_GET_VALUE
.ACL_32_3                      ANOP
.*                               AIF      (&GET_VALUE_ERR)
                                AIF      (NOT(&GET_VALUE_ERR)).AIF_52_1
&MSG                            SETC 'INVALID VALUE'
.*                               ACALL ERR_MSG
&ACALL_3_ERR_MSG             SETA      28
                                AGO      .ACL_3_ERR_MSG
.ACL_3_28                    ANOP
.*                               AEXIT AENTRY
                                AGO      .ACL_31_E
.*                               AEND
.AIF_52_1                      ANOP
.*                               ACALL SET_VAL_BLK
&ACALL_33_SET_VAL_BLK          SETA      2
                                AGO      .ACL_33_SET_VAL_BLK

```

zstrmac.txt

```

.ACCL_33_2          ANOP
.*                AEND
                  AGO    .ACS_1_E
.AC_S_1_G          ANOP
                  AGO
(C2A('&REC'(&IS_PARM,1))+1-64).ACS_1_2,.ACS_1_X,X
.AC_S_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.AX
CS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_1,.ACSX
_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1X
_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_XX
,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.X
ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACX
S_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_X,.ACS_1_3
                  AGO    .ACS_1_X
.AC_S_1_E          ANOP
.*                AEND
                  AGO    .AWH_8_T
.AC_S_1_E          ANOP
.*                AIF    (&VALUE_CNT EQ 0)
                  AIF    (NOT(&VALUE_CNT EQ 0)).AIF_53_1
&MSG              SETC 'NO AWHEN VALUES FOUND'
.*                ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA    29
                  AGO    .ACL_3_ERR_MSG
.ACCL_3_29        ANOP
.*                AEND
.ACIF_53_1        ANOP
.*                AEND
.ACCL_31_E        ANOP
                  AGO    (&ACALL_31_PROC_AWHEN_VALUES).ACL_31_1
.ACCL_31_SKIP    ANOP
.*
.* SET_VAL_BLK    AWHEN BLOCK NUMBER FOR VALUE
.*
.*                AENTRY SET_VAL_BLK
                  AGO    .ACL_33_SKIP
.ACCL_33_SET_VAL_BLK ANOP
.*                AIF    (&VALUE LT &LVL_ACASE_FIRST(&LVL))
                  AIF    (NOT(&VALUE LT &LVL_ACASE_FIRST(&LVL))).AIF_54_1

```

zstrmac.txt

```
&LVL_ACASE_FIRST(&LVL) SETA &VALUE
.*      AEND
.AIF_54_1 ANOP
.*      AIF  (&VALUE GT &LVL_ACASE_LAST(&LVL))
          AIF  (NOT(&VALUE GT &LVL_ACASE_LAST(&LVL))).AIF_55_1
&LVL_ACASE_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'
.*      ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA 30
          AGO .ACL_3_ERR_MSG
.ACL_3_30 ANOP
.*      AEND
.AIF_56_1 ANOP
&(&VAL_BLK)(&INDEX) SETA &LVL_BCNT(&LVL) SET BLK # FOR VAL
.*      AEND
          AGO (&ACALL_33_SET_VAL_BLK).ACL_33_1,.ACL_33_2
.ACL_33_SKIP ANOP
.*
.* GET_VALUE - DEC, C'?', OR X'??'
.*
.*      AENTRY GET_VALUE
          AGO .ACL_32_SKIP
.ACL_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)
```

```

                                zstrmac.txt
&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 (UPPER '&REC'(&IS_PARM,1) EQ 'C') RPI 911
                                AGO .AIF_57_E
.AIF_57_1 ANOP
                                AIF (NOT(UPPER '&REC'(&IS_PARM,1) EQ 'C')).AIF_57_2
RPIX
                                911
.*
                                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 (UPPER '&REC'(&IS_PARM,1) EQ 'X') RPI 911
                                AGO .AIF_57_E
.AIF_57_2 ANOP

```

```

                                zstrmac.txt
AIF      (NOT(UPPER '&REC'(&IS_PARM,1) EQ 'X')).AIF_57_3
RPIX
        911
.*      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

```

zstrmac.txt

```

.AIF_63_1 ANOP
&GET_VALUE_ERR SETB 1
.*      AEND
.AIF_63_E ANOP
.*      AEND
          AGO    (&ACALL_32_GET_VALUE).ACL_32_1,.ACL_32_2,.ACL_32_3
.ACL_32_SKIP ANOP
.*
.* AWHILE - GEN LABELD AIF TO END
.*
.*      AENTRY PROC_AWHILE
          AGO    .ACL_14_SKIP
.ACL_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,*)
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      32
          AGO    .ACL_19_PUNCH_REC
.ACL_19_32 ANOP
&PCH_REC SETC  '.AWH &LVL_TCNT(&LVL)_T'
.*      ACALL PUNCH_LAB
&ACALL_22_PUNCH_LAB SETA      17
          AGO    .ACL_22_PUNCH_LAB
.ACL_22_17 ANOP
&GEN_AIF_TRUE SETB 0          GEN BRANCH IF FALSE
&GEN_AIF_TAG SETC 'E'
.*      ACALL GEN_AIF
&ACALL_23_GEN_AIF SETA      4
          AGO    .ACL_23_GEN_AIF
.ACL_23_4 ANOP
.*      AIF    (&GEN_AIF_ERR)
          AIF    (NOT(&GEN_AIF_ERR)).AIF_65_1
&MSG          SETC 'AWHILE EXPRESSION ERROR'
.*      ACALL ERR_MSG
&ACALL_3_ERR_MSG SETA      31
          AGO    .ACL_3_ERR_MSG
.ACL_3_31      ANOP
.*      AELSE
          AGO    .AIF_65_E
.AIF_65_1 ANOP
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA      33

```

zstrmac.txt

```
                AGO    .ACL_19_PUNCH_REC
.ACL_19_33      ANOP
.*             AEND
.AIF_65_E      ANOP
.*             AEND
                AGO    (&ACALL_14_PROC_AWHILE).ACL_14_1
.ACL_14_SKIP   ANOP
.*
.* FIND_NAME OPERAND AND SET ACALL_INDEX TO EXISTING OR NEW ENTRY
.* SET FIND_NAME_ERR IF PARM ERROR
.*
.*             AENTRY FIND_NAME
                AGO    .ACL_29_SKIP
.ACL_29_FIND_NAME ANOP
&FIND_NAME_ERR SETB 0
.*             ACALL FIND_PARM
&ACALL_18_FIND_PARM SETA    4
                AGO    .ACL_18_FIND_PARM
.ACL_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')
&ACALL_INDEX   SETA 1
.*             AWHILE (&ACALL_INDEX LE &ACALL_NAME_TOT)
.AWH_11_T      ANOP
                AIF    (NOT(&ACALL_INDEX LE
&ACALL_NAME_TOT)).AWH_11_E
.*             AIF    ('&ACALL_NAME(&ACALL_INDEX)' EQ '&NAME')
                AIF    (NOT('&ACALL_NAME(&ACALL_INDEX)' EQ
'&NAME')X
                ).AIF_67_1
.*             AEXIT AENTRY  EXIT WITH ACALL_INDEX SET
                AGO    .ACL_29_E
.*             AEND
.AIF_67_1      ANOP
&ACALL_INDEX   SETA &ACALL_INDEX+1
.*             AEND
                AGO    .AWH_11_T
.AWH_11_E      ANOP
.*             AIF    (&ACALL_INDEX GT &ACALL_NAME_TOT)
                AIF    (NOT(&ACALL_INDEX GT &ACALL_NAME_TOT)).AIF_68_1
```

zstrmac.txt

```
&ACALL_NAME_TOT      SETA &ACALL_INDEX
&ACALL_NAME(&ACALL_INDEX) SETC '&NAME'
.*                   AEND
.AIF_68_1            ANOP
.*                   AEND
.AIF_66_E           ANOP
.*                   AEND
.ACL_29_E           ANOP
                    AGO   (&ACALL_29_FIND_NAME).ACL_29_1,.ACL_29_2
.ACL_29_SKIP        ANOP
.*
.* FIND_PARM OPERAND TERMINATED WITH SPACE
.* SET FIND_PARM_ERR IF ERROR
.*
.*                   AENTRY FIND_PARM
                    AGO   .ACL_18_SKIP
.ACL_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   .ACL_18_E
.*                   AEND
.AIF_69_1          ANOP
&IS_PARM           SETA &IS_PARM+1
.*                   AEND
                    AGO   .AWH_12_T
.AWH_12_E          ANOP
```

zstrmac.txt

```

&FIND_PARM_ERR SETB 1
.*      AEND
.ACCL_18_E ANOP
        AGO
(&ACALL_18_FIND_PARM).ACL_18_1,.ACL_18_2,.ACL_18_3,.ACX
        L_18_4
.ACCL_18_SKIP ANOP
.*
.* PUNCH LABEL WITH ANOP ALIGNED WITH AOP IF POSSIBLE
.*
.*      AENTRY PUNCH_LAB
        AGO      .ACL_22_SKIP
.ACCL_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'
.*      ACALL PUNCH_REC
&ACALL_19_PUNCH_REC SETA  34
        AGO      .ACL_19_PUNCH_REC
.ACCL_19_34 ANOP
.*      AEND
        AGO
(&ACALL_22_PUNCH_LAB).ACL_22_1,.ACL_22_2,.ACL_22_3,.ACX
L_22_4,.ACL_22_5,.ACL_22_6,.ACL_22_7,.ACL_22_8,.ACL_22_9X
, .ACL_22_10,.ACL_22_11,.ACL_22_12,.ACL_22_13,.ACL_22_14,X
        .ACL_22_15,.ACL_22_16,.ACL_22_17
.ACCL_22_SKIP ANOP
.*
.* PUNCH &PCH_REC WITH CONTINUATION FORMATTING AND RETURN TO CALLER
.* BASED ON &PUNCH_REC
.*
.*      AENTRY PUNCH_REC
        AGO      .ACL_19_SKIP
.ACCL_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

```

```

                                zstrmac.txt
.*          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
(&ACALL_19_PUNCH_REC).ACL_19_1,.ACL_19_2,.ACL_19_3,.ACX
L_19_4,.ACL_19_5,.ACL_19_6,.ACL_19_7,.ACL_19_8,.ACL_19_9X
,.ACL_19_10,.ACL_19_11,.ACL_19_12,.ACL_19_13,.ACL_19_14,X
.ACL_19_15,.ACL_19_16,.ACL_19_17,.ACL_19_18,.ACL_19_19,.X
ACL_19_20,.ACL_19_21,.ACL_19_22,.ACL_19_23,.ACL_19_24,.AX
CL_19_25,.ACL_19_26,.ACL_19_27,.ACL_19_28,.ACL_19_29,.ACX
            L_19_30,.ACL_19_31,.ACL_19_32,.ACL_19_33,.ACL_19_34
.ACL_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

```

zstrmac.txt

```
      AGO      .ACL_23_SKIP
.ACCL_23_GEN_AIF ANOP
&GEN_AIF_ERR SETB 0
.*          ACALL FIND_EXP
&ACALL_30_FIND_EXP SETA      2
      AGO      .ACL_30_FIND_EXP
.ACCL_30_2 ANOP
.*          AIF      (&FIND_EXP_ERR)
      AIF      (NOT(&FIND_EXP_ERR)).AIF_74_1
&GEN_AIF_ERR SETB 1
.*          AEXIT AENTRY
      AGO      .ACL_23_E
.*          AEND
.ACIF_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
.ACIF_75_1 ANOP
&PCH_REC     SETC    '&OP&EXP&LAB'
.*          AEND
.ACIF_75_E 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
.ACIF_76_1 ANOP
.*          AEND
.ACCL_23_E ANOP
      AGO
(&ACALL_23_GEN_AIF).ACL_23_1,.ACL_23_2,.ACL_23_3,.ACL_X
      23_4
.ACCL_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      .ACL_30_SKIP
```

zstrmac.txt

```

.ACL_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 .ACL_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
.ACL_30_E ANOP
        AGO (&ACALL_30_FIND_EXP).ACL_30_1,.ACL_30_2
.ACL_30_SKIP ANOP
.*
.* ERR_MSG ISSUE ERROR MESSAGE AND COUNT ERRORS
.*
.*      AENTRY ERR_MSG
        AGO .ACL_3_SKIP
.ACL_3_ERR_MSG ANOP
&ERRORS SETA &ERRORS+1
        MNOTE 8,'ZSTRMAC ERROR &MSG AT LINE &LINE'

```

zstrmac.txt

```
      PUNCH  ' MNOTE 8''ZSTRMAC ERROR &MSG',DDNAME=SYSUT2
.*      AEND
      AGO
(&ACALL_3_ERR_MSG).ACL_3_1,.ACL_3_2,.ACL_3_3,.ACL_3_4,X
.ACL_3_5,.ACL_3_6,.ACL_3_7,.ACL_3_8,.ACL_3_9,.ACL_3_10,.X
ACL_3_11,.ACL_3_12,.ACL_3_13,.ACL_3_14,.ACL_3_15,.ACL_3_X
16,.ACL_3_17,.ACL_3_18,.ACL_3_19,.ACL_3_20,.ACL_3_21,.ACX
L_3_22,.ACL_3_23,.ACL_3_24,.ACL_3_25,.ACL_3_26,.ACL_3_27X
      ,.ACL_3_28,.ACL_3_29,.ACL_3_30,.ACL_3_31
.ACL_3_SKIP ANOP
      MEND
      ZSTRMAC
      END
```