L 12

```
   SSSSSSSSSSSSS   YYY          YYY      SSSSSSSSSSSSS
   SSSSSSSSSSSSS   YYY          YYY      SSSSSSSSSSSSS
   SSSSSSSSSSSSS   YYY          YYY      SSSSSSSSSSSSS
SSS                YYY          YYY   SSS
SSS                YYY          YYY   SSS
SSS                YYY          YYY   SSS
SSS                   YYY    YYY      SSS
SSS                   YYY    YYY      SSS
SSS                   YYY    YYY      SSS
   SSSSSSSSS             YYY             SSSSSSSSS
   SSSSSSSSS             YYY             SSSSSSSSS
   SSSSSSSSS             YYY             SSSSSSSSS
             SSS         YYY                       SSS
             SSS         YYY                       SSS
             SSS         YYY                       SSS
             SSS         YYY                       SSS
             SSS         YYY                       SSS
             SSS         YYY                       SSS
SSSSSSSSSSSSS            YYY          SSSSSSSSSSSSS
SSSSSSSSSSSSS            YYY          SSSSSSSSSSSSS
SSSSSSSSSSSSS            YYY          SSSSSSSSSSSSS
```

FILEID**SYSASCEFC

```
 SSSSSSSS   YY      YY   SSSSSSSS     AAAAAA     SSSSSSSS     CCCCCCCC  EEEEEEEEEE  FFFFFFFFFF    CCCCCCCC
 SSSSSSSS   YY      YY   SSSSSSSS     AAAAAA     SSSSSSSS     CCCCCCCC  EEEEEEEEEE  FFFFFFFFFF    CCCCCCCC
SS          YY      YY  SS          AA      AA  SS          CC          EE          FF          CC
SS          YY      YY  SS          AA      AA  SS          CC          EE          FF          CC
SS            YY  YY    SS          AA      AA  SS          CC          EE          FF          CC
SS            YY  YY    SS          AA      AA  SS          CC          EE          FF          CC
  SSSSSS        YY        SSSSSS    AA      AA    SSSSSS    CC          EEEEEEEE    FFFFFFFF    CC
  SSSSSS        YY        SSSSSS    AA      AA    SSSSSS    CC          EEEEEEEE    FFFFFFFF    CC
        SS      YY              SS  AAAAAAAAAA          SS  CC          EE          FF          CC
        SS      YY              SS  AAAAAAAAAA          SS  CC          EE          FF          CC
        SS      YY              SS  AA      AA          SS  CC          EE          FF          CC              ....
        SS      YY              SS  AA      AA          SS  CC          EE          FF          CC              ....
SSSSSSSS        YY        SSSSSSSS  AA      AA  SSSSSSSS      CCCCCCCC  EEEEEEEEEE  FF            CCCCCCCC      ....
SSSSSSSS        YY        SSSSSSSS  AA      AA  SSSSSSSS      CCCCCCCC  EEEEEEEEEE  FF            CCCCCCCC      ....

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

SYS
V04

Table of contents

```
0000          .TITLE  SYSASCEFC - ASSOCIATE, DISASSOCIATE AND DELETE COMMON EVENT FLAG CLU
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;*****************************************************************************
0000    ;*                                                                           *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    ;*  ALL RIGHTS RESERVED.                                                     *
0000    ;*                                                                           *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    ;*  TRANSFERRED.                                                             *
0000    ;*                                                                           *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    ;*  CORPORATION.                                                             *
0000    ;*                                                                           *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    ;*                                                                           *
0000    ;*                                                                           *
0000    ;*****************************************************************************
0000
0000    ;++
0000    ; FACILITY:     EXECUTIVE, SYSTEM SERVICES
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ; ENVIRONMENT:
0000    ;
0000    ;--
0000    ;
0000          .PAGE
0000          .SBTTL  HISTORY                 ; DETAILED
0000    ;
0000    ; AUTHOR:       R. HUSTVEDT     CREATION DATE: 3-OCT-76
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-001 KDM42758        Kathleen D. Morse       05-Jan-1983
0000    ;               Add logic to implement the port quota for shared memory
0000    ;               common event flag clusters.  Change GET_REFCNT_LCK to the
0000    ;               global, EXE$CEBREFLCK, for use by the SYSGEN SHARE command.
0000    ;               Change EXE$SHM_DELETE to EXE$SHMCEBDEL for use by SYSGEN, also.
0000    ;
0000    ;
0000    ;
0000    ;       ,_: VERSION
0000    ; 01     -
0000
```

G 12

```
                   0000          .SBTTL  DECLARATIONS
                   0000
                   0000  ;
                   0000  ; INCLUDE FILES:
                   0000  ;
                   0000
                   0000          $CEBDEF                         ; DEFINE COMMON EVENT CONTROL BLOCK
                   0000          $DYNDEF                         ; DEFINE DATA STRUCTURE TYPES
                   0000          $IPLDEF                         ; DEFINE IPL VALUES
                   0000          $JIBDEF                         ; DEFINE JIB OFFSETS
                   0000          $PCBDEF                         ; DEFINE PCB OFFSETS
                   0000          $PRDEF                          ; DEFINE PROCESSOR REGISTERS
                   0000          $PRQDEF                         ; DEFINE INTER-PROCESSOR REQUEST
                   0000          $PRVDEF                         ; DEFINE PRIVILEGE BITS
                   0000          $SHBDEF                         ; DEFINE SHARED MEMORY CONTROL BLOCK
                   0000          $SHDDEF                         ; DEFINE SHARED MEMORY COMMON DATA PAGE
                   0000          $SSDEF                          ; DEFINE STATUS CODE VALUES
                   0000          $STATEDEF                       ; DEFINE SCHEDULER STATES
                   0000
                   0000  ;
                   0000  ; EQUATED SYMBOLS:
                   0000  ;
                   0000
         00000004  0000  EFN=4                                   ; EFN ARGUMENT DISPLACEMENT
         00000008  0000  NAME=8                                  ; NAME ARGUMENT DISPLACEMENT
         0000000C  0000  PROT=12                                 ; PROTECTION MASK DISPLACEMENT
         00000010  0000  PERM=16                                 ; PERMANENT INDICATOR DISPLACEMENT
                   0000
                   0000  ;
                   0000  ; OWN STORAGE:
                   0000  ;
                   0000
```

```
                          0000          .SBTTL  RETQUOTA - RETURN CEF QUOTA
                          0000    ;+
                          0000    ; FUNCTIONAL DESCRIPTION:
                          0000    ;       RETQUOTA RETURNS THE QUOTA TO THE ORIGINAL CREATOR OF THE
                          0000    ;       COMMON EVENT CLUSTER.  IF THE PROCESS IDENTIFIED AS THE CLUSTER
                          0000    ;       CREATOR HAS BEEN DELETED, THEN NO OPERATION IS PERFORMED.
                          0000    ;
                          0000    ; CALLING SEQUENCE:
                          0000    ;       BSB/JSB RETQUOTA
                          0000    ;
                          0000    ; INPUT PARAMETERS:
                          0000    ;       R0 - ADDRESS OF COMMON EVENT BLOCK
                          0000    ;       R2 - PROCESS INDEX OF CREATOR
                          0000    ;
                          0000    ; OUTPUT PARAMETERS:
                          0000    ;       R0 - R0 (PRESERVED)
                          0000    ;
                          0000    ;--
                          0000
                          0000    RETQUOTA:                                       ; RETURN QUOTA
                          0000            DSBINT  #IPL$_SYNCH                     ; BLOCK SYSTEM EVENTS
   51  0000'DF42     D0   0006            MOVL    @W^SCH$GL_PCBVEC[R2],R1         ; CONVERT TO PCB ADDRESS
   0C A0    60 A1    D1   000C            CMPL    PCB$L_PID(R1),CEB$L_PID(R0)     ; CHECK FOR EXISTENCE
                08   12   0011            BNEQ    10$                             ; NO, IGNORE QUOTA
   51    0080 C1     D0   0013            MOVL    PCB$L_JIB(R1),R1                ; GET JIB ADDRESS
             34 A1   B6   0018            INCW    JIB$W_TQCNT(R1)                 ; GIVE BACK QUOTA
                          001B    10$:    ENBINT                                  ; ENABLE SYSTEM EVENTS
                     05   001E            RSB                                     ; RETURN
```

I 12

```
                    001F        .SBTTL  EXE$DACEFC - DISASSOCIATE COMMON EVENT CLUSTER
                    001F
                    001F  ;++
                    001F  ; FUNCTIONAL DESCRIPTION:
                    001F  ;         EXE$DACEFC IMPLEMENTS THE DISASSOCIATE COMMON EVENT CLUSTER
                    001F  ;         SYSTEM SERVICE.  THE ISSUING PROCESS IS DISASSOCIATED WITH
                    001F  ;         THE CLUSTER SELECTED BY THE SPECIFIED EVENT FLAG NUMBER AND
                    001F  ;         THE CLUSTER CONTROL BLOCK (CEB) WILL BE RELEASED IF THE REFERENCE
                    001F  ;         COUNT BECOMES ZERO AND IT IS NOT MARKED PERMANENT.
                    001F  ;
                    001F  ; CALLING SEQUENCE:
                    001F  ;         CALLG   ARGLIST,EXE$DACEFC
                    001F  ;
                    001F  ; INPUT PARAMETERS:
                    001F  ;         EFN(AP) - EVENT FLAG NUMBER TO SELECT CLUSTER TO DISASSOCIATE.
                    001F  ;         R4 - PCB ADDRESS OF CURRENT PROCESS
                    001F  ;
                    001F  ; OUTPUT PARAMETERS:
                    001F  ;         R0 - COMPLETION STATUS
                    001F  ;
                    001F  ; COMPLETION CODES:
                    001F  ;         SS$_NORMAL  -  SUCCESSFUL COMPLETION
                    001F  ;         SS$_ILLEFC  -  ILLEGAL EVENT FLAG NUMBER FOR COMMON EVENT CLUSTER
                    001F  ;                        MUST BE 64-127
                    001F  ;
                    001F  ; SIDE EFFECTS:
                    001F  ;         IF THE REFERENCE COUNT FOR THE SELECTED COMMON EVENT CLUSTER
                    001F  ;         BECOMES ZERO AND IT IS NOT MARKED AS PERMANENT, IT WILL BE
                    001F  ;         DELETED.
                    001F  ;
                    001F  ;--
                    001F
                00000000        .PSECT  Y$EXEPAGED,BYTE
                    0000  EXE$DACEFC::                                  ; DISASSOCIATE COMMON EVENT CLUSTER
            0FFC    0000        .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; SAVE REGISTERS R2-R11
         11   10    0002        BSBB    GETCLUSTER                    ; GET CLUSTER INDEX AND VALIDATE
    58 A448   DD    0004        PUSHL   PCB$L_EFC2P(R4)[R8]           ; PUSH ADDRESS OF CEB FOR THIS CLUSTER
    58 A448   D4    0008        CLRL    PCB$L_EFC2P(R4)[R8]           ; CLEAR POINTER
       026D   30    000C        BSBW    CEBLOCK                       ; LOCK MUTEX FOR WRITE
    57   54   D0    000F        MOVL    R4,R7                         ; REMEMBER PCB ADDRESS
       017F   31    0012        BRW     DISASOC                       ; DISASSOCIATE CLUSTER
```

```
                              0015          .SBTTL  GETCLUSTER - GET AND VERIFY CLUSTER NUMBER
                              0015
                              0015    ;++
                              0015    ; FUNCTIONAL DESCRIPTION:
                              0015    ;       GETCLUSTER EXTRACTS THE CLUSTER NUMBER FROM THE EFN ARGUMENT,
                              0015    ;       VALIDATES IT AND SUBTRACTS THE LOCAL EVENT CLUSTER BIAS TO
                              0015    ;       FORM A COMMON EVENT CLUSTER INDEX.  IF THE CLUSTER NUMBER
                              0015    ;       IS NOT A VALID COMMON EVENT CLUSTER, AN EXIT IS MADE FROM
                              0015    ;       THE SERVICE BY ISSUING A RET.
                              0015    ;
                              0015    ; CALLING SEQUENCE:
                              0015    ;       BSB/JSB GETCLUSTER
                              0015    ;
                              0015    ; INPUT PARAMETERS:
                              0015    ;       EFN(AP) - EVENT FLAG NUMBER
                              0015    ;
                              0015    ; OUTPUT PARAMETERS:
                              0015    ;       R8 - COMMON EVENT CLUSTER INDEX
                              0015    ;
                              0015    ; COMPLETION CODES:     (ON ERROR RETURN ONLY)
                              0015    ;       SS$_ILLEFC - ILLEGAL COMMON EVENT CLUSTER
                              0015    ;
                              0015    ;--
                              0015
                              0015    GETCLUSTER:                                   ; GET COMMON EVENT CLUSTER INDEX
  58   04 AC   03   05   EE   0015          EXTV    #5,#3,EFN(AP),R8              ; EXTRACT CLUSTER FROM EFN
               58   02   C2   001B          SUBL    #2,R8                         ; SUBTRACT BIAS
                    01   19   001E          BLSS    10$                           ; ILLEGAL CLUSTER NUMBER
                         05   0020          RSB                                   ; SUCCESSFUL RETURN
                              0021
        50   00EC 8F     3C   0021    10$:  MOVZWL  #SS$_ILLEFC,R0                ; SET ERROR CODE
                         04   0026          RET                                   ; AND EXIT SERVICE CALL
                              0027
                              0027
```

```
                  0027        .SBTTL  EXE$ASCEFC - ASSOCIATE AND CREATE COMMON EVENT CLUSTER
                  0027
                  0027  ;++
                  0027  ; FUNCTIONAL DESCRIPTION:
                  0027  ;       EXE$ASCEFC IMPLEMENTS THE ASSOCIATE COMMON EVENT CLUSTER SYSTEM
                  0027  ;       SERVICE.  IF THE NAMED COMMON EVENT CLUSTER DOES NOT ALREADY
                  0027  ;       EXIST, IT WILL BE CREATED.  IF IT ALREADY EXISTS, THE PROCESS
                  0027  ;       WILL BE ASSOCIATED WITH IT, PROTECTION PERMITTING.
                  0027  ;
                  0027  ; CALLING SEQUENCE:
                  0027  ;       CALLG   ARGLIST,EXE$ASCEFC
                  0027  ;
                  0027  ; INPUT PARAMETERS:
                  0027  ;       EFN(AP)  - EVENT FLAG NUMBER TO SPECIFY CLUSTER NUMBER
                  0027  ;       NAME(AP) - ADDRESS OF STRING DESCRIPTOR FOR NAME OF CLUSTER
                  0027  ;       PROT(AP) - PROTECTION MASK FOR CLUSTER IF CREATING
                  0027  ;       PERM(AP) - INDICATOR TO SPECIFY CREATION OF PERMANENT CLUSTER
                  0027  ;       R4 - PCB ADDRESS
                  0027  ;
                  0027  ; IMPLICIT INPUTS:
                  0027  ;       PCB OF CURRENT PROCESS
                  0027  ;
                  0027  ; OUTPUT PARAMETERS:
                  0027  ;       R0 - COMPLETION STATUS CODE
                  0027  ;
                  0027  ; IMPLICIT OUTPUTS:
                  0027  ;       PCB$L_EFC2P - POINTER TO COMMON EVENT CONTROL BLOCK IF CLUSTER 2
                  0027  ;       PCB$L_EFC3P - POINTER TO COMMON EVENT CONTROL BLOCK IF CLUSTER 3
                  0027  ;
                  0027  ; COMPLETION CODES:
                  0027  ;       SS$_NORMAL      - SUCCESSFUL COMPLETION
                  0027  ;       SS$_ACCVIO      - LOGICAL NAME INACCESSIBLE
                  0027  ;       SS$_EXPORTQUOTA - NO PORT QUOTA
                  0027  ;       SS$_EXQUOTA     - QUOTA FOR COMMON EVENT BLOCK CREATION EXHAUSTED
                  0027  ;       SS$_ILLEFC      - ILLEGAL CLUSTER NUMBER (NOT 0,1,2,3)
                  0027  ;       SS$_IVLOGNAM    - COUNT FOR LOGICAL NAME NOT 1-15
                  0027  ;       SS$_NOPRIV      - PRIVILEGE VIOLATION
                  0027  ;       SS$_SHMNOTCNCT  - SHARED MEMORY NOT CONNECTED
                  0027  ;       SS$_TOOMANYLNAM - TOO MANY LOGICAL NAME TRANSLATIONS
                  0027  ;
                  0027  ; SIDE EFFECTS:
                  0027  ;       AN IMPLIED DISASSOCIATE IS PERFORMED ON THE SPECIFIED CLUSTER
                  0027  ;       AND THE PREVIOUSLY ASSOCIATED COMMON EVENT CLUSTER MAY BE
                  0027  ;       DELETED IF IT IS NOT MARKED PERMANENT AND ITS REFERENCE COUNT
                  0027  ;       BECOMES ZERO.
                  0027  ;
                  0027  ;--
                  0027
                  0027        .ENABL  LSB
                  0027  EXE$ASCEFC::                                ; ASSOCIATE COMMON EVENT CLUSTER
            0FFC  0027        .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTER SAVE MASK
         EA   10  0029        BSBB    GETCLUSTER                  ; GET CLUSTER INDEX AND VALIDATE
  59  08 AC   D0  002B        MOVL    NAME(AP),R9                 ; GET POINTER TO CLUSTER NAME
      10 AC   DD  002F        PUSHL   PERM(AP)                    ; SAVE PERMANENT INDICATOR
    5E   20   C2  0032        SUBL    #<4*8>,SP                   ; ALLOCATE SPACE FOR NAMES ON STACK
       0251   30  0035        BSBW    LOCLUST                     ; LOCATE CLUSTER
  5A 0080 C7  D0  0038        MOVL    PCB$L_JIB(R7),R10           ; GET ADDRESS OF JIB
```

```
5A   34 AA    9E 003D              MOVAB   JIB$W_TQCNT(R10),R10     ; POINT TO TQCNT FIELD
                 0041      ;
                 0041      ; R2 = ADR OF LOCAL COMMON EVENT BLOCK (PERHAPS SLAVE)
                 0041      ;      (0 IF NOT YET CREATED)
                 0041      ; R4 = ADR OF SHARED MEMORY CONTROL BLOCK (IF IN SHARED MEMORY)
                 0041      ; R5 = ADR OF SHARED MEMORY COMMON DATA PAGE (IF IN SHARED MEMORY)
                 0041      ; R6 = ADR OF SHARED MEMORY MASTER COMMON EVENT BLOCK (IF IN SHARED MEMORY)
                 0041      ;      (0 IF NOT YET CREATED)
                 0041      ; R7 = ADR OF PROCESS CONTROL BLOCK
                 0041      ; R8 = INDEX TO APPROPRIATE CLUSTER IN PCB
                 0041      ; R10 = ADDRESS OF JIB$W_TQCNT, QUOTA FOR COMMON EVENT BLOCKS
                 0041      ;  0(SP) = ASCIC COMMON EVENT FLAG CLUSTER NAME
                 0041      ; 16(SP) = ASCIC SHARED MEMORY NAME
                 0041      ; 32(SP) = PERMANENT INDICATOR
                 0041      ;
          52  D5 0041              TSTL    R2                       ; WAS NAME FOUND
          03  13 0043              BEQL    10$                      ; BR IF NAME NOT FOUND, CREATE CEB
        0140  31 0045              BRW     ASSOCIATE                ; BR AS NAME FOUND, ASSOCIATE IT
          56  D5 0048      10$:    TSTL    R6                       ; IS THERE A MASTER SHMCEB?
          05  13 004A              BEQL    11$                      ; BR IF NO MASTER EXISTS
20 AE   0B A6 90 004C              MOVB    CEB$B_STS(R6),32(SP)     ; FORCE SLAVES TO BE SAME AS MASTER
   0A 20 AE   E9 0051      11$:    BLBC    32(SP),15$               ; SKIP PRMCEB PRIV CHK IF CREATING TEMP
05 6C B7 0A   E0 0055              BBS     #PRV$V_PRMCEB,@PCB$L_PHD(R7),15$ ;BR IF HAVE PRIV TO CREATE
     50  24   9A 005A      12$:    MOVZBL  #SS$_NOPRIV,R0           ; SET ERROR CODE
          38  11 005D              BRB     36$                      ; AND RETURN FROM SYSTEM SERVICE
       10 AE  95 005F      15$:    TSTB    16(SP)                   ; IS THIS A SHARED MEMORY CLUSTER?
          11  12 0062              BNEQ    30$                      ; BR IF YES, DONT CHARGE QUOTA
    1C 20 AE  E8 0064              BLBS    32(SP),32$               ; DONT CHARGE QUOTA IF PERM CLUSTER
          6A  B5 0068              TSTW    (R10)                    ; CHECK IF QUOTA REMAINING
          05  1A 006A              BGTRU   20$                      ; CONTINUE IF QUOTA REMAINING
     50  1C   9A 006C              MOVZBL  #SS$_EXQUOTA,R0          ; SET STATUS CODE FOR QUOTA EXHAUSTED
          3A  11 006F              BRB     37$                      ; AND RETURN FROM SYSTEM SERVICE
          6A  B7 0071      20$:    DECW    (R10)                    ; ACCOUNT FOR COMMON EVENT BLOCK
          0F  11 0073              BRB     32$                      ; GO ALLOCATE A LOCAL MEMORY CEB
          56  D5 0075      30$:    TSTL    R6                       ; IS MASTER SHMCEB ALREADY CREATED?
          0B  12 0077              BNEQ    32$                      ; BR IF MASTER SHMCEB EXISTS
DC 6C B7  1B  E1 0079              BBC     #PRV$V_SHMEM,@PCB$L_PHD(R7),12$ ; EXIT SERVICE IF NOT PRIVILEGED
        0317  30 007E              BSBW    ALOSHMCEB                ; GO ALLOCATE A MASTER SHMCEB
                 0081                                               ; R6 = ADR OF MASTER SHMCEB
       27 50  E9 0081              BLBC    R0,37$                   ; BR IF NO FREE CEB
       FF79'  30 0084      32$:    BSBW    EXE$ALLOCCEB             ; ALLOCATE SPACE FOR CEB/SLAVCEB
                 0087                                               ; R2 = ADR OF SLAVE CEB
       24 50  E8 0087              BLBS    R0,40$                   ; CONTINUE IF SUCCESSFUL
          56  D5 008A              TSTL    R6                       ; IS THIS A SHARED MEMORY CLUSTER?
          09  12 008C              BNEQ    36$                      ; BR IF YES, DONT RETURN QUOTA
    19 20 AE  E8 008E              BLBS    32(SP),37$               ; SKIP QUOTA IF CREATING PERM
          6A  B6 0092      35$:    INCW    (R10)                    ; RETURN QUOTA IF ERROR
        0166  31 0094              BRW     EXIT                     ; RETURN ERROR TO SYSTEM SERVICE
          56  D5 0097      36$:    TSTL    R6                       ; IS THERE A MASTER CEB?
          10  13 0099              BEQL    37$                      ; BR ON NO MASTER CEB
          50  DD 009B              PUSHL   R0                       ; REMEMBER FIRST ERROR CODE
        03A5  30 009D              BSBW    DEC_SHMCEB_REF           ; RELEASE LOCK PREVENTING DELETION
       50 8ED0   00A0              POPL    R0                       ; IGNORE POTENTIAL INTERLOCKING PROBLEM
   04 66  00  E0 00A3              BBS     #CEB$V_VALID,CEB$L_CEBFL(R6),37$ ; BR IF MASTER NOT JUST CREATED
   00 66  01  E7 00A7              BBCCI   #CEB$V_LOCKED,CEB$L_CEBFL(R6),37$ ; RELEASE MASTER FOR OTHER USE
        014F  31 00AB      37$:    BRW     EXIT                     ; EXIT IF NO SPACE FOR CEB
                 00AE      40$:                                     ;
```

```
                                  00AE   ;
                                  00AE   ; R2 = ADR OF LOCAL COMMON EVENT BLOCK (PERHAPS SLAVE)
                                  00AE   ; R4 = ADR OF SHARED MEMORY CONTROL BLOCK (IF IN SHARED MEMORY)
                                  00AE   ; R5 = ADR OF SHARED MEMORY COMMON DATA PAGE (IF IN SHARED MEMORY)
                                  00AE   ; R6 = ADR OF SHARED MEMORY MASTER COMMON EVENT BLOCK (IF IN SHARED MEMORY)
                                  00AE   ; R7 = ADR OF PROCESS CONTROL BLOCK
                                  00AE   ; R10 = ADDRESS OF JIB$W_TQCNT, QUOTA FOR COMMON EVENT BLOCK
                                  00AE   ;
                                  00AE   ;
                                  00AE   ; FIRST INITIALIZE THE CEB FIELDS USED IN THE LOCAL MEMORY CEB AND NOT
                                  00AE   ; IN THE SLAVE CEB.  THIS ALLOWS A BRANCH BACK TO 41$ IF A RACE OCCURS
                                  00AE   ; DURING CREATION OF THE MASTER CEB.
                                  00AE   ;
          24 A2     0C AC    B0   00AE             MOVW    PROT(AP),CEB$W_PROT(R2) ; SET PROTECTION MASK
 0B A2   01   01    20 AE    F0   00B3             INSV    32(SP),#CEB$V_PERM,#1,CEB$B_STS(R2) ; SET PERMANENT FLAG
                                  00BA   ;
                                  00BA   ; NOW INITIALIZE THE CEB FIELDS IN COMMON TO A LOCAL MEMORY CEB AND A
                                  00BA   ; SLAVE CEB.
                                  00BA   ;
          14 A2     14 A2    DE   00BA   41$:      MOVAL   CEB$L_WQFL(R2),CEB$L_WQFL(R2) ; SET UP WAIT QUEUE HEADER
          18 A2     14 A2    DE   00BF             MOVAL   CEB$L_WQFL(R2),CEB$L_WQBL(R2) ; FORWARDS AND BACKWARDS LINKS
                    1C A2    B4   00C4             CLRW    CEB$W_WQCNT(R2)          ; CLEAR QUEUE COUNT
                    10 A2    D4   00C7             CLRL    CEB$L_EFC(R2)            ; EVENT FLAG SET
                    26 A2    B4   00CA             CLRW    CEB$W_REFC(R2)           ; INIT REFERENCE COUNT TO 1
        20 A2     00BC C7    D0   00CD             MOVL    PCB$L_UIC(R7),CEB$L_UIC(R2) ; SET UIC OF CLUSTER OWNER
          0C A2     20 AA    D0   00D3             MOVL    <JIB$L_MPID-JIB$W_TQCNT>(R10),CEB$L_PID(R2) ; AND MASTER PID SO
                                  00D8                                              ; THAT QUOTA WILL BE RETURNED CORRECTLY
                       3C    BB   00D8             PUSHR   #^M<R2,R3,R4,R5>         ; SAVE REGISTERS FOR MOVC3
     28 A2    10 AE    10    28   00DA             MOVC3   #16,16(SP),CEB$T_EFCNAM(R2) ; FILL IN CLUSTER NAME
                       3C    BA   00E0             POPR    #^M<R2,R3,R4,R5>         ; RESTORE REGISTERS
          1E A2        03    B0   00E2             MOVW    #SCH$C_CEF,CEB$W_STATE(R2) ; SET WAIT STATE NUMBER
                    10 AE    95   00E6             TSTB    16(SP)                   ; IS THIS IN SHARED MEMORY?
                       03    12   00E9             BNEQ    39$                      ; BR IF IN SHARED MEMORY
                     008D    31   00EB             BRW     50$                      ; BR IF IN LOCAL MEMORY
          51        15 A4    9A   00EE   39$:      MOVZBL  SHB$B_PORT(R4),R1        ; GET PROCESSOR PORT INDEX
          0A A2        2D    90   00F2             MOVB    #DYN$C_SLAVCEB,CEB$B_TYPE(R2) ; INDICATE ENTRY IS SLAVE CEB
          10 A2     10 A6    D0   00F6             MOVL    CEB$L_EFC(R6),CEB$L_EFC(R2) ; COPY FLAGS FROM MASTER CEB
          38 A2        54    D0   00FB             MOVL    R4,CEB$L_SHB(R2)         ; SET ADR OF SHARED MEMORY CONTROL BLOCK
         38 A641       52    D0   00FF             MOVL    R2,CEB$L_VASLAVE1(R6)[R1] ; SET VA OF SLAVE CEB
          40 A2        56    D0   0104             MOVL    R6,CEB$L_MASTER(R2)      ; SET ADR OF MASTER IN SLAVE CEB
             0315            30   0108             BSBW    INC_SHMCEB_REF           ; ADD REFCNT FOR SLAVCEB TO MASTER CEB
                                  010B   ;         BLBC    R0,XXX                   ; BR IF INTERLOCK ERROR
       00 0B A2        01    E5   010B             BBCC    #CEB$V_PERM,CEB$B_STS(R2),115$ ; SLAVES ARE ALWAYS TEMP
    51    08 A5        55    C1   0110   115$:     ADDL3   R5,SHD$L_CEFPTR(R5),R1   ; GET ADR OF SHM CEF TABLE
       51     56       51    C3   0115             SUBL3   R1,R6,R1                 ; GET OFFSET TO CEB ENTRY
          50        08 A6    3C   0119             MOVZWL  CEB$W_SIZE(R6),R0        ; GET SIZE OF ONE ENTRY
                 51    50    C6   011D             DIVL2   R0,R1                    ; GET # OF ENTRIES INTO TABLE
          3C A2        51    B0   0120             MOVW    R1,CEB$W_INDX(R2)        ; SET INDEX INTO MASTER CEB TABLE
          03 66        00    E1   0124             BBC     #CEB$V_VALID,CEB$L_CEBFL(R6),112$ ; BR IF MASTER JUST CREATED
                     0050    31   0128             BRW     50$                      ; BR IF MASTER NOT JUST CREATED
                                  012B   ;
                                  012B   ; HERE CHECK IF THE NEWLY CREATED MASTER SHMGSD IS UNIQUE OR NOT.
                                  012B   ;
                     034C    30   012B   112$:     BSBW    UNIQUE_CEB               ; CHECK THAT ANOTHER CEB WAS NOT CREATED
                       5B    D5   012E             TSTL    R11                      ; WAS THERE A DUPLICATE?
                       41    13   0130             BEQL    48$                      ; BR ON NO DUPLICATE
          53        15 A4    9A   0132             MOVZBL  SHB$B_PORT(R4),R3        ; GET PORT # INDEX
```

```
   00 0B A6     01   E6  0136          BBSSI   #CEB$V_PERM,CEB$B_STS(R6),113$ ; INHIBIT DELETION CODE EXECUTION
      38 A643        D4  013B  113$:   CLRL    CEB$L_VASLAVE1(R6)[R3]   ; INDICATE NO SLAVE ASSOCIATED
        0303         30  013F          BSBW    DEC_SRMCEB_REF           ; RELEASE ASSOC AND SLAVECEB REFCNTS ON
        0300         30  0142          BSBW    DEC_SHMCEB_REF           ; MASTER CEB JUST CREATED
                         0145                                           ; IGNORE INTERLOCK ERRORS FOR BOTH
      00 66     01   E7  0145          BBCCI   #CEB$V_LOCKED,CEB$L_CEBFL(R6),42$ ; RELEASE CEB FOR OTHER USE
      38 AB43        D5  0149  42$:    TSTL    CEB$L_VASLAVE1(R11)[R3]  ; IS THERE ALREADY A SLAVE CEB?
          14         13  014D          BEQL    43$                      ; BR TO USE SLAVE CEB JUST CREATED
      50    52       D0  014F          MOVL    R2,R0                    ; SET ADR OF SLAVE CEB JUST CREATED
     00000000'EF     16  0152          JSB     EXE$DEANONPAGED          ; RELEASE SLAVE CEB
   53    15 A4       9A  0158          MOVZBL  SHB$B_PORT(R4),R3        ; GET PORT # INDEX (DESTROYED BY JSB)
 52    38 AB43       D0  015C          MOVL    CEB$L_VASLAVE1(R11)[R3],R2 ; GET ADR OF OLD SLAVE CEB
          25         11  0161          BRB     ASSOCIATE                ; GO ASSOCIATE WITH OLD SLAVE CEB
      56    5B       D0  0163  43$:    MOVL    R11,R6                   ; SET ADR OF DUPLICATE MASTER CEB
   0B A2    0B A6    90  0166          MOVB    CEB$B_STS(R6),CEB$B_STS(R2) ; SET SLAVE STATUS SAME AS MASTER
   24 A2    24 A6    B0  016B          MOVW    CEB$W_PROT(R6),CEB$Q_PROT(R2) ; COPY PROTECTION FROM DUPLICATE
        FF47         31  0170          BRW     41$                      ; GO INITIALIZE SLAVE CEB AGAIN
      00 66     00   E6  0173  48$:    BBSSI   #CEB$V_VALID,CEB$L_CEBFL(R6),49$ ; INDICATE ENTRY IS VALID
      00 66     01   E7  0177  49$:    BBCCI   #CEB$V_LOCKED,CEB$[_CEBFL(R6),50$ ; CLEAR THE LOCK BIT
                         017B  ;
                         017B  ; ADD NEW SLAVE CEB TO LOCAL MEMORY DATA STRUCTURES.
                         017B  ;
     00000000'EF     B6  017B  50$:    INCW    SCH$GW_CEBCNT            ; INCREMENT COUNT OF COMMON EVENT BLOCKS
 00000000'EF    62   0E  0181          INSQUE  (R2),SCH$GQ_CEBHD        ; INSERT CEB IN LIST
                         0188
                         0188          .DSABL  LSB
                         0188
                         0188  ASSOCIATE:                               ; R6 MAY NOT BE MASTER CEB FOR R2 HERE
      26 A2          B6  0188          INCW    CEB$W_REFC(R2)           ; REFERENCE CLUSTER
      58 A748        DD  018B          PUSHL   PCB$L_EFC2P(R7)[R8]      ; GET POINTER TO CURRENT CLUSTER
 58 A748    52       D0  018F          MOVL    R2,PCB$L_EFC2P(R7)[R8]   ; SET CLUSTER POINTER IN PCB
                         0194  DISASOC:
                         0194          .ENABL  LSB
      50    8E       D0  0194          MOVL    (SP)+,R0                 ; GET OLD CLUSTER ADDRESS
            61       13  0197          BEQL    EXITN                    ; NONE
   0A A0    2D       91  0199          CMPB    #DYN$C_SLAVCEB,CEB$B_TYPE(R0) ; IS THIS A SLAVE CEB?
            14       12  019D          BNEQ    5$                       ; BR IF NORMAL LOCAL MEMORY CEB
   54    38 A0       D0  019F          MOVL    CEB$L_SHB(R0),R4         ; GET ADR OF SHARED MEMORY CONTROL BLOCK
   55    04 A4       D0  01A3          MOVL    SHB$L_DATAPAGE(R4),R5    ; GET ADR OF SHARED MEMORY DATA PAGE
   56    40 A0       D0  01A7          MOVL    CEB$L_MASTER(R0),R6      ; GET ADR OF MASTER CEB
            50       DD  01AB          PUSHL   R0                       ; SAVE ADR OF SLAVE CEB
          0295       30  01AD          BSBW    DEC_SHMCEB_REF           ; DECREMENT ASSOC REFCNT IN MASTER CEB
            50     8ED0  01B0          POPL    R0                       ; IGNORE POTENTIAL INTERLOCK ERROR
      26 A0          B7  01B3  5$:     DECW    CEB$W_REFC(R0)           ; DECREMENT REFERENCE COUNT
            42       14  01B6          BGTR    EXITN                    ; STILL BUSY, CANT RELEASE YET
   0A A0    2D       91  01B8          CMPB    #DYN$C_SLAVCEB,CEB$B_TYPE(R0) ; IS THIS A SLAVE CEB?
            14       13  01BC          BEQL    10$                      ; BR IF YES, DONT RETURN QUOTA
 37 0B A0   01       E0  01BE          BBS     #CEB$V_PERM,CEB$B_STS(R0),EXITN ; PERMANENT CLUSTER
 0A 0B A0   00       E0  01C3          BBS     #CEB$V_NOQUOTA,CEB$B_STS(R0),10$ ; CHECK FOR QUOTA
   52    0C A0       3C  01C8          MOVZWL  CEB$L_PID(R0),R2         ; GET PIX OF CREATOR
     00000000'EF     16  01CC          JSB     RETQUOTA                 ; RETURN QUOTA
     00000000'EF     B7  01D2  10$:    DECW    SCH$GW_CEBCNT            ; DECREASE COUNT OF COMMON EVENT BLOCKS
      51    60       0F  01D8          REMQUE  (R0),R1                  ; REMOVE FROM LIST
   0A A0    2D       91  01DB          CMPB    #DYN$C_SLAVCEB,CEB$B_TYPE(R0) ; IS THIS A SLAVE CEB?
            13       12  01DF          BNEQ    20$                      ; BR IF NORMAL LOCAL MEMORY CEB
     00000000'EF     16  01E1          JSB     EXE$DEANONPAGED          ; RELEASE SLAVE CEB
   51    15 A4       9A  01E7          MOVZBL  SHB$B_PORT(R4),R1        ; PORT # FOR THIS PROCESSOR
```

```
        38 A641   D4  01EB   424          CLRL    CEB$L_VASLAVE1(R6)[R1]  ; INDICATE NO SLAVE CEB FOR PROCESSOR
           0253   30  01EF   425          BSBW    DEC_SRMCEB_REF          ; RELEASE SLAVCEB REFCNT IN MASTER CEB
             09   11  01F2   426          BRB     EXIT                    ; BRANCH TO COMMON CODE
                      01F4   427
     00000000'EF  16  01F4   428 20$:     JSB     EXE$DEANONPAGED         ; DEALLOCATE SPACE FOR LOCAL CEB
          50  01  9A  01FA   429 EXITN:   MOVZBL  #SS$_NORMAL,R0          ; SET NORMAL COMPLETION STATUS
             50   DD  01FD   430 EXIT:    PUSHL   R0                      ; SAVE COMPLETION STATUS
  50 00000000'EF  DE  01FF   431 EXITE:   MOVAL   EXE$GL_CEBMTX,R0        ; GET ADDRESS OF CEB MUTEX
          54  57  D0  0206   432          MOVL    R7,R4                   ; RESTORE ADDRESS OF PCB
     00000000'EF  16  0209   433          JSB     SCH$UNLOCK              ; AND FREE IT
             01   BA  020F   434          POPR    #^M<R0>                 ; RESTORE COMPLETION CODE
                      0211   435          SETIPL  #0                      ; ENABLE
                  04  0214   436          RET                             ; AND RETURN TO CALLER
                      0215   437
                      0215   438          .DSABL  LSB
```

```
                      0215         .SBTTL  EXE$DLCEFC - DELETE COMMON EVENT CLUSTER
         00000004     0215   NAME=4                                      ; NAME DISPLACEMENT IN ARGLIST
                      0215
                      0215
                      0215   ;++
                      0215   ; FUNCTIONAL DESCRIPTION:
                      0215   ;       EXE$DLCEFC IMPLEMENTS THE DELETE COMMON EVENT CLUSTER SYSTEM
                      0215   ;       SERVICE WHICH PERMITS A PERMANENT COMMON EVENT CLUSTER TO BE
                      0215   ;       MARKED FOR DELETION WHEN ITS REFERENCE COUNT BECOMES ZERO.
                      0215   ;
                      0215   ; CALLING SEQUENCE:
                      0215   ;       CALLG   ARGLIST,EXE$DLCEFC
                      0215   ;
                      0215   ; INPUT PARAMETERS:
                      0215   ;       NAME(AP) - ADDRESS OF STRING DESCRIPTOR FOR NAME OF CLUSTER
                      0215   ;
                      0215   ; OUTPUT PARAMETERS:
                      0215   ;       R0 - COMPLETION STATUS
                      0215   ;
                      0215   ; COMPLETION CODES:
                      0215   ;       SS$_NORMAL  -  SUCCESSFUL COMPLETION OF SERVICE
                      0215   ;       SS$_NOPRIV  -  INSUFFICIENT PRIVILEGE TO DELETE CLUSTER
                      0215   ;
                      0215   ; SIDE EFFECTS:
                      0215   ;       IF THE REFERENCE COUNT FOR THE SPECIFIED CLUSTER IS ZERO,
                      0215   ;       THE CLEARING OF THE PERMANENT BIT WILL CAUSE IT TO BE DELETED.
                      0215   ;
                      0215   ;--
                      0215
                      0215   EXE$DLCEFC::                                ; DELETE COMMON EVENT CLUSTER
               0FFC   0215          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTER SAVE MASK
    59    04 AC  D0   0217          MOVL    NAME(AP),R9                 ; GET POINTER TO CLUSTER NAME
       5E    20  C2   021B          SUBL    #32,SP                      ; CREATE SPACE FOR NAME STRINGS
             69  10   021E          BSBB    LOCLUST                     ; LOCATE CLUSTER BY NAME
             52  D5   0220          TSTL    R2                          ; WAS NAME FOUND
             37  13   0222          BEQL    40$                         ; NO, GO CHECK IF THERE'S A MASTER CEB
 20 A2   00BC C7  D1  0224          CMPL    PCB$L_UIC(R7),CEB$L_UIC(R2) ; OWNER OF CLUSTER?
             0A  13   022A          BEQL    10$                         ; YES, PERMIT ANYWAY
   05 6C B7  0A  E0   022C          BBS     #PRV$V_PRMCEB,@PCB$L_PHD(R7),10$ ; CONTINUE IF PRIVILEGED
       50    24  9A   0231   5$:    MOVZBL  #SS$_NOPRIV,R0              ; SET ERROR CODE
             C7  11   0234          BRB     EXIT                        ; EXIT RELEASING MUTEX
   05 0B A2  01  E5   0236   10$:   BBCC    #CEB$V_PERM,CEB$B_STS(R2),20$ ; CLEAR PERMANENT FLAG
   00 0B A2  00  E2   023B          BBSS    #CEB$V_NOQUOTA,CEB$B_STS(R2),20$ ; SET NO QUOTA
             52  DD   0240   20$:   PUSHL   R2                          ; SAVE CLUSTER ADDRESS
          26 A2  B6   0242          INCW    CEB$W_REFC(R2)              ; TEMPORARILY REFERENCE IT
             56  D5   0245          TSTL    R6                          ; IS THERE A MASTER CEB, ALSO?
             0F  13   0247          BEQL    30$                         ; BR IF NO MASTER CEB TO DELETE
   E3 6C B7  1B  E1   0249          BBC     #PRV$V_SHMEM,@PCB$L_PHD(R7),5$ ; BR IF NOT PRIV TO DELETE
   00 0B A6  01  E7   024E          BBCCI   #CEB$V_PERM,CEB$B_STS(R6),25$ ; CHANGE STATUS TO TEMP FOR DELET
 1F A6    15 A4  90   0253   25$:   MOVB    SHB$B_PORT(R4),CEB$B_DELETPORT(R6) ; INDICATE PORT # OF DELETOR
           FF39  31   0258   30$:   BRW     DISASOC                     ; DISASSOCIATE
                      025B                                              ; RELEASING CEB IF IDLE
             56  D5   025B   40$:   TSTL    R6                          ; IS THERE A MASTER CEB TO DELETE?
             9B  13   025D          BEQL    EXITN                       ; BR IF NONE, EXIT WITH NORMAL STATUS
 20 A6   00BC C7  D1  025F          CMPL    PCB$L_UIC(R7),CEB$L_UIC(R6) ; OWNER OF CLUSTER?
             05  13   0265          BEQL    50$                         ; YES, PERMIT DELETION
   C5 6C B7  0A  E1   0267          BBC     #PRV$V_PRMCEB,@PCB$L_PHD(R7),5$ ; BR IF NOT PRIVILEGED
```

```
C0 6C B7    1B  E1  026C   497 50$:   BBC    #PRV$V_SHMEM,@PCB$L_PHD(R7),5$ ; BR IF NOT PRIV TO DELETE
00 0B A6    01  E7  0271   498 60$:   BBCCI  #CEB$V_PERM,CEB$B_STS(R6),70$ ; CHANGE STATUS TO TEMP FOR DELETE
          01CC  30  0276   499 70$:   BSBW   DEC_SHMCEB_REF          ; RELEASE SEARCH CNT; TRY TO DEL MASTER
          FF81  31  0279   500        BRW    EXIT                    ; EXIT AT COMPLETION CODE REPORTING
                    027C   501
                    027C   502
                    027C   503
```

```
                          027C          .SBTTL  CEBLOCK - LOCK COMMON EVEN BLOCK MUTEX FOR WRITE
                          027C
                          027C  ;++
                          027C  ; FUNCTIONAL DESCRIPTION:
                          027C  ;       CEBLOCK LOCKS THE COMMON EVENT BLOCK MUTEX FOR WRITE AND
                          027C  ;       RETURNS WITH IPL AT IPL$_ASTDEL.
                          027C  ;
                          027C  ; CALLING SEQUENCE:
                          027C  ;       BSB/JSB CEBLOCK
                          027C  ;
                          027C  ; INPUT PARAMETERS:
                          027C  ;
                          027C  ;       R4 - PCB ADDRESS
                          027C  ;
                          027C  ;--
                          027C
                          027C  CEBLOCK:                                        ; LOCK CEB MUTEX FOR WRITE
50   00000000'EF    DE    027C          MOVAL   EXE$GL_CEBMTX,R0                ; SET ADDRESS OF MUTEX
     00000000'EF    17    0283          JMP     SCH$LOCKW                       ; LOCK IT FOR WRITE
                          0289
```

```
                0289        .SBTTL  LOCLUST - LOCATE CLUSTER GIVEN NAME
                0289
      00000008  0289  NAME=8                               ; NAME DISPLACEMENT IN ARGLIST
                0289  ;++
                0289  ; FUNCTIONAL DESCRIPTION:
                0289  ;       LOCLUST SEARCHES THE LIST OF COMMON EVENT BLOCKS FOR THE
                0289  ;       NAMED CLUSTER AND RETURNS THE ADDRESS OF THE CLUSTER.
                0289  ;       IF NO ERROR CONDITIONS ARE ENCOUNTERED, LOCLUST RETURNS WITH
                0289  ;       THE CEB MUTEX LOCKED FOR WRITE.  IN THE EVENT OF ERROR, THE
                0289  ;       MUTEX IS UNLOCKED AND A RET IS ISSUED TO EXIT THE SERVICE.
                0289  ;
                0289  ; CALLING SEQUENCE:
                0289  ;       BSB/JSB LOCLUST
                0289  ;
                0289  ; INPUT PARAMETERS:
                0289  ;       04(SP) - 16 BYTE AREA INTO WHICH CEF CLUSTER NAME WILL BE MOVED
                0289  ;       14(SP) - 16 BYTE AREA INTO WHICH SHARED MEMORY NAME WILL BE MOVED
                0289  ;       R4 - PCB ADDRESS
                0289  ;       R9 - ADDRESS OF NAME STRING DESCRIPTOR
                0289  ;
                0289  ; OUTPUT PARAMETERS:
                0289  ;       R0 - COMPLETION STATUS
                0289  ; ALL CASES:
                0289  ;       R1,R3 - DESTROYED
                0289  ;       R7 - PCB ADDRESS OF CURRENT PROCESS
                0289  ; IF LOCAL MEMORY COMMON EVENT CLUSTER NAME:
                0289  ;       R2 - ADDRESS OF LOCAL MEMORY CEB  (0 IF NAME NOT FOUND)
                0289  ;       R4 - R5 DESTROYED
                0289  ;       R6 - 0
                0289  ; IF SHARED MEMORY COMMON EVENT CLUSTER NAME:
                0289  ;       R2 - ADDRESS OF SLAVE CEB (0 IF NO SLAVE FOUND)
                0289  ;       R4 - SHARED MEMORY CONTROL BLOCK ADDRESS
                0289  ;       R5 - SHARED MEMORY COMMON DATA PAGE ADDRESS
                0289  ;       R6 - ADDRESS OF SHARED MEMORY MASTER CEB (0 IF NAME NOT FOUND)
                0289  ;
                0289  ; COMPLETION CODES:
                0289  ;
                0289  ;       SS$_ACCVIO       - LOGICAL NAME INACCESSIBLE
                0289  ;       SS$_IVLOGNAM     - COUNT FOR LOGICAL NAME NOT 1-15
                0289  ;       SS$_NOPRIV       - PRIVILEGE VIOLATION
                0289  ;       SS$_SHMNOTCNCT   - SHARED MEMORY NOT CONNECTED
                0289  ;       SS$_TOOMANYLNAM  - TOO MANY LOGICAL NAME TRANSLATIONS
                0289  ;
                0289  ;--
                0289
                0289  LOCLUST:                             ; LOCATE CLUSTER
                0289        .ENABL  LSB
                0289
   0E00 8F  BB  0289        PUSHR   #^M<R9,R10,R11>        ; SAVE REGISTERS
   57   54  D0  028D        MOVL    R4,R7                  ; SAVE PCB ADDRESS
                0290        IFNORD  #8,(R9),30$            ; TEST DESCRIPTOR READABILITY
                0296  ;
                0296  ; CREATE STRING DESCRIPTORS FOR COMMON EVENT CLUSTER AND SHARED MEMORY NAME
                0296  ; BUFFERS.  THEN PERFORM LOGICAL TRANSLATION OF USER-PROVIDED NAME TO
                0296  ; DETERMINE ACTUAL CLUSTER NAME AND SHARED MEMORY NAME.
                0296  ;
   51   69  9A  0296        MOVZBL  (R9),R1                ; GET COUNT FROM DESCRIPTOR (LE 255)
```

```
                  7A    13  0299          BEQL    50$                             ; ILLEGAL STRING LENGTH
                            029B          IFNORD  R1,@4(R9),30$                   ; CHECK READABILITY OF STRING
               11 AE    9F  02A2          PUSHAB  <<4*4>+1>(SP)                   ; SET UP STRING DESCRIPTOR FOR
                  0F    DD  02A5          PUSHL   #15                             ; CEF CLUSTER NAME
             5B   5E    D0  02A7          MOVL    SP,R11                          ; SET ADR OF CEF NAME STR DSC
               29 AE    9F  02AA          PUSHAB  <<4*10>+1>(SP)                  ; SET UP STRING DESCRIPTOR FOR
                  0F    DD  02AD          PUSHL   #15                             ; SHARED MEMORY NAME
             5A   5E    D0  02AF          MOVL    SP,R10                          ; SET ADR OF SHM NAME STR DSC
               FD4B'    30  02B2          BSBW    MMG$CEFTRNLOG                   ; DETERMINE CEF AND SHM NAMES
               62 50    E9  02B5          BLBC    R0,60$                          ; BR IF ERROR IN LOGICAL NAME TRANS
         18 AB    6B    90  02B8          MOVB    (R11),<4*6>(R11)                ; MAKE ASCIC CEF NAME
                  57    13  02BC          BEQL    50$                             ; BR IF NO CEF NAME SPECIFIED
               04 AB    D7  02BE          DECL    4(R11)                          ; CONVERT TO POINT TO ASCIC NAME
         30 AA    6A    90  02C1          MOVB    (R10),<4*12>(R10)               ; MAKE ASCIC SHM NAME
                  63    12  02C5          BNEQ    FIND_SHM_CEB                    ; BR IF CEF CLUSTER IS IN SHARED MEMORY
                  B3    10  02C7          BSBB    CEBLOCK                         ; LOCK CEB LIST TO PROHIBIT WRITERS
                            02C9      ;
                            02C9      ; FIND COMMON EVENT BLOCK IN LOCAL MEMORY.
                            02C9      ;
   5A   00000000'EF     DE  02C9          MOVAL   SCH$GQ_CEBHD,R10                ; GET ADDRESS OF CEB LIST
             56   5A    D0  02D0          MOVL    R10,R6                          ; AND POINTER TO HEAD OF LIST
             56   66    D0  02D3  10$:    MOVL    (R6),R6                         ; FLINK ONWARD
             56   5A    D1  02D6          CMPL    R10,R6                          ; CHECK FOR END OF LIST
                  40    13  02D9          BEQL    15$                             ; BR IF AT END OF LIST, ENTRY NOT FOUND
   00BE C7    22 A6     B1  02DB  40$:    CMPW    CEB$W_GRP(R6),PCB$W_GRP(R7) ; SAME GROUP?
                  F0    12  02E1          BNEQ    10$                             ; NO, TRY ANOTHER
 28 A6    04 BB     10  29  02E3          CMPC3   #16,@4(R11),CEB$T_EFCNAM(R6) ; COMPARE THE NAMES
                  E8    12  02E9          BNEQ    10$                             ; NOT EQUAL, TRY ANOTHER
         0A A6    2D    91  02EB          CMPB    #DYN$C_SLAVCEB,CEB$B_TYPE(R6) ; SLAVE CEB?
                  E2    13  02EF          BEQL    10$                             ; BR IF SLAVE, DON'T USE CEB
   00BC C7    20 A6     D1  02F1          CMPL    CEB$L_UIC(R6),PCB$L_UIC(R7) ; OWNER UIC?
                  24    13  02F7          BEQL    20$                             ; YES CONTINUE
            20 24 A6    E9  02F9          BLBC    CEB$W_PROT(R6),20$              ; CONTINUE IF NO ERROR
                            02FD      ;
                            02FD      ; NO ACESSS ALLOWED TO COMMON EVENT BLOCK.
                            02FD      ;
             54   57    D0  02FD  45$:    MOVL    R7,R4                           ; RESET ADR OF PCB
   50   00000000'EF     DE  0300          MOVAL   EXE$GL_CEBMTX,R0                ; SET ADDRESS OF MUTEX
        00000000'EF     16  0307          JSB     SCH$UNLOCK                      ; UNLOCK MUTEX
             50   24    9A  030D          MOVZBL  #SS$_NOPRIV,R0                  ; SET ERROR CODE
                        04  0310          RET                                     ; AND RETURN FROM SERVICE
             50   0C    3C  0311  30$:    MOVZWL  #SS$_ACCVIO,R0                  ; INDICATE ACCESS VIOLATION
                        04  0314          RET                                     ; AND EXIT SERVICE
        50    0154 8F   3C  0315  50$:    MOVZWL  #SS$_IVLOGNAM,R0                ; INDICATE ILLEGAL STRING
                        04  031A  60$:    RET                                     ; EXIT FROM SYSTEM SERVICE
                            031B      ;
                            031B      ; RETURN ADDRESS OF LOCAL COMMON EVENT BLOCK (IF FOUND) ELSE 0.
                            031B      ;
                  56    D4  031B  15$:    CLRL    R6                              ; INDICATE NONE FOUND
             52   56    D0  031D  20$:    MOVL    R6,R2                           ; RETURN ADR OF CEB
                  56    D4  0320          CLRL    R6                              ; RETURN NO MASTER SHMGSD
             5E   10    C0  0322  25$:    ADDL    #<4*4>,SP                       ; CLEAN STR DSC'S OFF STACK
             0E00 8F    BA  0325          POPR    #^M<R9,R10,R11>                 ; RESTORE REGISTERS
                        05  0329          RSB                                     ; AND RETURN
                            032A
                            032A      ;
                            032A      ; FIND COMMON EVENT BLOCK IN SHARED MEMORY.
```

```
                        032A          ;
                        032A          FIND_SHM_CEB:
                        032A                                                     ; R10=ADR OF STR DESC FOR SHM NAME
            04 AA   D7  032A                  DECL    4(R10)                     ; POINT TO ASCIC SHM NAME
            FCD0'   30  032D                  BSBW    MMG$FINDSHB                ; FIND SHARED MEMORY CONTROL BLOCK
            E7 50   E9  0330                  BLBC    R0,60$                     ; BR IF MEMORY IS NOT CONNECTED
                        0333                                                     ; R4=SHB ADR, R5=SHD ADR
         59    54   D0  0333                  MOVL    R4,R9                      ; REMEMBER SHB ADR
         54    57   D0  0336                  MOVL    R7,R4                      ; SET CURRENT PCB ADR
             FF40   30  0339                  BSBW    CEBLOCK                    ; LOCK CEB LIST TO PROHIBIT WRITERS
         54    59   D0  033C                  MOVL    R9,R4                      ; RESTORE SHB ADR
   56    55 08 A5   C1  033F                  ADDL3   SHD$L_CEFPTR(R5),R5,R6     ; ADR OF 1ST CEB IN SHM CEF TABLE
         59 08 A6   3C  0344                  MOVZWL  CEB$W_SIZE(R6),R9          ; GET SIZE OF ONE CEB
         5A 1C A5   3C  0348                  MOVZWL  SHD$W_CEFMAX(R5),R10       ; GET MAXIMUM CEB'S IN TABLE
               38   13  034C                  BEQL    160$                       ; BR IF NO CEB TBL AT ALL
             00CF   30  034E  140$:           BSBW    INC_SHMCEB_REF             ; LOCK THIS CEB AGAINST DELETION
            2C 50   E9  0351                  BLBC    R0,155$                    ; DONT USE ENTRY IF IT IS REFCNT LOCKED
         25 66 01   E0  0354                  BBS     #CEB$V_LOCKED,CEB$L_CEBFL(R6),150$ ; BR IF ENTRY BEING CREATED
         21 66 00   E1  0358                  BBC     #CEB$V_VALID,CEB$L_CEBFL(R6),150$ ; BR IF ENTRY NOT IN USE
   00BE C7 22 A6    B1  035C                  CMPW    CEB$W_GRP(R6),PCB$W_GRP(R7) ; IS THIS CEB IN PROCESS' GROUP?
               19   12  0362                  BNEQ    150$                       ; BR IF IT IS IN DIFFERENT GROUP
 28 A6    04 BB 10  29  0364                  CMPC3   #16,@4(R11),CEB$T_EFCNAM(R6) ; DOES NAME OF CLUSTER MATCH?
               11   12  036A                  BNEQ    150$                       ; BR IF DIFFERENT NAME
   00BC C7 20 A6    D1  036C                  CMPL    CEB$L_UIC(R6),PCB$L_UIC(R7) ; DOES PROCESS OWN THIS CLUSTER?
               18   13  0372                  BEQL    170$                       ; BR IF PROCESS OWNS THIS CLUSTER
         14 24 A6   E9  0374                  BLBC    CEB$W_PROT(R6),170$        ; BR IF CLUSTER IS NOT PROTECTED
             00CA   30  0378                  BSBW    DEC_SHMCEB_REF             ; RELEASE LOCK AGAINST DELETION
                        037B                                                     ; IGNORE POTENTIAL INTERLOCK ERROR
               80   11  037B                  BRB     45$                        ; GO RELEASE MUTEX AND EXIT SERVICE
             00C5   30  037D  150$:           BSBW    DEC_SHMCEB_REF             ; RELEASE LOCK AGAINST DELETION
                        0380                                                     ; IGNORE POTENTIAL INTERLOCK ERROR
         56    59   C0  0380  155$:           ADDL2   R9,R6                      ; ADR OF NEXT CEB IN SHM CEF TABLE
         C8    5A   F5  0383                  SOBGTR  R10,140$                   ; SEARCH ALL OF CEB TABLE
               56   D4  0386  160$:           CLRL    R6                         ; INDICATE NO MASTER CEB FOUND
               52   D4  0388                  CLRL    R2                         ; INDICATE NO SLAVE CEB FOUND
               09   11  038A                  BRB     180$                       ; RETURN TO COMMON CODE
         52 15 A4   9A  038C  170$:           MOVZBL  SHB$B_PORT(R4),R2          ; GET PORT # INDEX FOR VA OF SLAVE CEB
   52    38 A642    D0  0390                  MOVL    CEB$L_VASLAVE1(R6)[R2],R2  ; SET ADR OF SLAVE CEB
             FF8A   31  0395  180$:           BRW     25$                        ; RETURN TO COMMON CODE
                        0398
                        0398                  .DSABL  LSB
```

```
                         0398          .SBTTL  ALOSHMCEB - ALLOCATE SHARED MEMORY MASTER COMMON EVENT BLOCK
                         0398
                         0398   ;++
                         0398   ; FUNCTIONAL DESCRIPTION:
                         0398   ;       THIS ROUTINE ALLOCATES AND INITIALIZES A SHARED MEMORY MASTER COMMON
                         0398   ;       EVENT BLOCK.  SOME FIELDS HAVE ALREADY BEEN INITIALIZED BY SYSGEN.
                         0398   ;
                         0398   ; CALLING SEQUENCE:
                         0398   ;       BSBW    ALOSHMCEB
                         0398   ;
                         0398   ; INPUT PARAMETERS:
                         0398   ;       R4 - ADDRESS OF SHARED MEMORY CONTROL BLOCK
                         0398   ;       R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
                         0398   ;
                         0398   ; IMPLICIT INPUTS:
                         0398   ;       COMMON DATA PAGE MUST BE INITIALIZED.
                         0398   ;
                         0398   ; OUTPUT PARAMETERS:
                         0398   ;       R0 - COMPLETION STATUS CODE
                         0398   ;       R6 - ADDRESS OF SHARED MEMORY MASTER CEB, IF SUCCESSFUL
                         0398   ;          - 0, IF ERROR
                         0398   ;
                         0398   ; IMPLICIT OUTPUTS:
                         0398   ;       THE MASTER CEB IS INITIALIZED.  ITS REF COUNT IS UP BY ONE TO LOCK IT.
                         0398   ;
                         0398   ; COMPLETION CODES:
                         0398   ;       SS$_EXPORTQUOTA - NO PORT QUOTA
                         0398   ;       SS$_NORMAL      - SUCCESSFUL COMPLETION
                         0398   ;       SS$_NOSHMBLOCK  - NO SHARED MEMORY MASTER CEB AVAILABLE
                         0398   ;
                         0398   ; SIDE EFFECTS:
                         0398   ;       THE MASTER CEB IS LOCKED PREVENTING OTHER PROCESSES FROM USING IT.
                         0398   ;
                         0398   ;--
                         0398
                         0398           .ENABL  LSB
                         0398   ALOSHMCEB:
                06  BB   0398           PUSHR   #^M<R1,R2>                      ; SAVE REGISTERS
         51  15 A4  9A   039A           MOVZBL  SHB$B_PORT(R4),R1               ; GET PORT # TO BE USED AS AN INDEX
7C A541    FFFF 8F  58   039E           ADAWI   #-1,SHD$W_CEFQUOTA(R5)[R1]      ; SUBTRACT PORT QUOTA FOR CEF
                67  19   03A5           BLSS    NO_QUOTA                        ; BR IF NO QUOTA AVAILABLE
   56    55  08 A5  C1   03A7           ADDL3   SHD$L_CEFPTR(R5),R5,R6          ; GET ADR OF FIRST CEB IN SHM TABLE
         52  08 A6  3C   03AC           MOVZWL  CEB$W_SIZE(R6),R2               ; GET SIZE OF ONE SHMCEB
         51  1C A5  3C   03B0           MOVZWL  SHD$W_CEFMAX(R5),R1             ; GET # OF ENTRIES IN TABLE
                0B  11   03B4           BRB     20$                             ; JOIN COMMON CODE
              008C  30   03B6   10$:    BSBW    DEC_SHMCEB_REF                  ; UNLOCK THE PREVIOUS ENTRY FOR DELETION
                         03B9                                                   ; IGNORE POTENTIAL INTERLOCK ERROR
            56  52  C0   03B9   15$:    ADDL2   R2,R6                           ; GET ADDRESS OF NEXT CEB IN TABLE
            02  51  F5   03BC           SOBGTR  R1,20$                          ; BR IF ANOTHER ENTRY TO CHECK
                46  13   03BF           BEQL    NO_FREE_CEB                     ; BR IF NO ENTRY AVAILABLE
              005C  30   03C1   20$:    BSBW    INC_SHMCEB_REF                  ; LOCK THE ENTRY
             F2 50  E9   03C4           BLBC    R0,15$                          ; DONT USE ENTRY IF IT IS REFCNT LOCKED
      EB 66     01  E0   03C7           BBS     #CEB$V_LOCKED,CEB$L_CEBFL(R6),10$ ; BR IF ENTRY BEING MODIFIED
      E7 66     00  E0   03CB           BBS     #CEB$V_VALID,CEB$L_CEBFL(R6),10$ ; BR IF ENTRY ALREADY IN USE
      E3 66     01  E6   03CF           BBSSI   #CEB$V_LOCKED,CEB$L_CEBFL(R6),10$ ; BR IF SOMEONE JUST TOOK IT
                         03D3   ;
                         03D3   ; FOUND AN UNUSED ENTRY.  NOW INITIALIZE IT.
```

```
                                03D3  ;
       1C A6   15 A4    90     03D3              MOVB    SHB$B_PORT(R4),CEB$B_LOCK(R6) ; INDICATE PORT LOCKING ENTRY
0B A6    01    01 2C AE  F0    03D8              INSV    44(SP),#CEB$V_PERM,#1,CEB$B_STS(R6) ; SET PERMANENT FLAG
               10 A6    D4     03DF              CLRL    CEB$L_EFC(R6)                 ; INITIALIZE EVENT FLAGS
       0C A6   60 A7    D0     03E2              MOVL    PCB$L_PID(R7),CEB$L_PID(R6) ; INITIALIZE UIC OF OWNER
     20 A6   00BC C7    D0     03E7              MOVL    PCB$L_UIC(R7),CEB$L_UIC(R6) ; AND UIC OF CLUSTER OWNER
       24 A6   0C AC    3C     03ED              MOVZWL  PROT(AP),CEB$W_PROT(R6) ; INITIALIZE PROTECTION OF CLUSTER
                  3C    BB     03F2              PUSHR   #^M<R2,R3,R4,R5>              ; SAVE REGISTERS
  28 A6    1C AE   10   28     03F4              MOVC3   #16,28(SP),CEB$T_EFCNAM(R6) ; SET NAME OF CLUSTER
                  3C    BA     03FA              POPR    #^M<R2,R3,R4,R5>              ; RESTORE REGISTERS
                               03FC              ASSUME  CEB$B_DELETPORT EQ <CEB$B_CREATPORT + 1>
       1E A6   15 A4    9B     03FC              MOVZBW  SHB$B_PORT(R4),CEB$B_CREATPORT(R6) ; INIT CREATOR/DELETOR PORTS
           50     01    9A     0401              MOVZBL  #SS$_NORMAL,R0                ; REPORT SUCCESS
                  06    BA     0404   30$:       POPR    #^M<R1,R2>                    ; RESTORE REGISTERS
                  05           0406              RSB                                   ; RETURN TO CREATE SLAVE CEB
                               0407
                               0407   NO_FREE_CEB:
     50    03B4 8F      3C     0407              MOVZWL  #SS$_NOSHMBLOCK,R0            ; REPORT FAILURE
                  05    11     040C              BRB     40$                           ; JOIN COMMON CODE
                               040E
                               040E   NO_QUOTA:                                        ; NO CEF PORT QUOTA
     50    03AC 8F      3C     040E              MOVZWL  #SS$_EXPORTQUOTA,R0           ; REPORT FAILURE
        51     15 A4    9A     0413   40$:       MOVZBL  SHB$B_PORT(R4),R1             ; GET PORT # TO BE USED AS AN INDEX
    7C A541       01    58     0417              ADAWI   #1,SHD$W_CEFQUOTA(R5)[R1] ; RETURN QUOTA ACQUIRED
                  56    D4     041C              CLRL    R6                            ; INDICATE NO MASTER CEB, TO USE COMMON
                               041E                                                    ; ERROR PATHS IN CALLER CODE
                  E4    11     041E              BRB     30$                           ; RETURN TO CLEAN UP
                               0420
                               0420              .DSABL  LSB
```

```
                      0420          .SBTTL  INC_SHMCEB_REF/DEC_SHMCEB_REF - ALTER REFCNT ON SHMEM MASTER CEB
                      0420
                      0420  ;++
                      0420  ; FUNCTIONAL DESCRIPTION:
                      0420  ;       THESE ROUTINES INCREMENT/DECREMENT RESPECTIVELY A PROCESSOR REFERENCE
                      0420  ;       COUNT IN A SHARED MEMORY COMMON EVENT BLOCK BY ONE.  THE PORT NUMBER
                      0420  ;       IS USED AS AN INDEX IN DETERMINING WHICH PROCESSOR REFERENCE COUNT TO
                      0420  ;       ALTER.  THIS IS DONE TO LOCK AN ENTRY SO THAT IT CANNOT BE RELEASED
                      0420  ;       WHILE A PROCESSOR IS SCANNING IT.  A REFERENCE LOCK BIT IN THE MASTER
                      0420  ;       MUST BE HELD WHILE UPDATING A REFERENCE COUNT AND WHILE CHECKING THAT
                      0420  ;       AN ENTRY'S REFERENCE COUNTS FOR DELETION.  THIS PREVENTS A RACE
                      0420  ;       CONDITION BETWEEN ASSOCIATION AND DELETION OF COMMON EVENT FLAGS.
                      0420  ;       NOTE THAT EXECUTION MUST ALREADY BE SINGLE STREAMED ON THE PROCESSOR
                      0420  ;       EXECUTING THIS CODE VIA MUTEXES.
                      0420  ;
                      0420  ;       THESE ROUTINES ASSUME THAT THE PROCESSOR REFERENCE COUNTS ARE
                      0420  ;       IMMEDIATELY PRECEEDED BY THE SLAVE CEB VIRTUAL ADDRESSES IN THE
                      0420  ;       MASTER CEB.
                      0420  ;
                      0420  ; CALLING SEQUENCE:
                      0420  ;       BSBW    INC_SHMCEB_REF/DEC_SHMCEBREF
                      0420  ;
                      0420  ; INPUT PARAMETERS:
                      0420  ;       R4 - ADDRESS OF SHARED MEMORY CONTROL BLOCK
                      0420  ;       R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE (USED BY EXE$SHMCEBDEL)
                      0420  ;       R6 - ADDRESS OF SHARED MEMORY MASTER COMMON EVENT BLOCK
                      0420  ;
                      0420  ; IMPLICIT INPUTS:
                      0420  ;       THE SHARED MEMORY MUST BE CONNECTED AND THE REFERENCE COUNTS IN THE
                      0420  ;       MASTER CEB SET UP AT SYSGEN TIME.
                      0420  ;
                      0420  ; OUTPUT PARAMETERS:
                      0420  ;       R0 - STATUS CODE
                      0420  ;
                      0420  ; IMPLICIT OUTPUTS:
                      0420  ;       THE MASTER CEB'S REF COUNT IS UP/DOWN BY ONE TO LOCK/RELEASE IT.
                      0420  ;
                      0420  ; COMPLETION CODES:
                      0420  ;       SS$_NORMAL - SUCCESSFUL COMPLETION
                      0420  ;       SS$_INTERLOCK - UNABLE TO OBTAIN ENTRY REF COUNT LOCK
                      0420  ;
                      0420  ; SIDE EFFECTS:
                      0420  ;       THE MASTER CEB CANNOT DISAPPEAR UNTIL THE REFERENCE COUNT IS RELEASED.
                      0420  ;
                      0420  ;--
                      0420
                      0420  INC_SHMCEB_REF:
        7E   51   7D  0420          MOVQ    R1,-(SP)                        ; SAVE REGISTERS
   51   15   A4   9A  0423          MOVZBL  SHB$B_PORT(R4),R1               ; GET PORT # FOR INDEX TO PROC REFCNT
   52   1D   A6   9A  0427          MOVZBL  CEB$B_PROCCNT(R6),R2            ; GET # OF PROCESSOR SLAVE VA'S IN CEB
52   52   02   78  042B          ASHL    #2,R2,R2                        ; CONVERT TO BYTE CNT, SLAVEVA=LONGWORD
        52   56   C0  042F          ADDL2   R6,R2                           ; OFFSET PAST THE SLAVE VA'S
           FC01'  30  0432          BSBW    EXE$CEBREFLCK                   ; ACQUIRE REF COUNT LOCK IN SHMCEB
        09   50   E9  0435          BLBC    R0,10$                          ; BR IF UNABLE TO ACQUIRE LOCK
38 A241   01   58  0438          ADAWI   #1,CEB$L_VASLAVE1(R2)[R1]       ; INCREMENT PROCESSOR REF COUNT
  00 66   02   E7  043D          BBCCI   #CEB$V_REFCNTLCK,CEB$L_CEBFL(R6),10$ ; RELEASE REF CNT LOCK
        51   8E   7D  0441  10$:    MOVQ    (SP)+,R1                        ; RESTORE REGISTER
```

```
                      05  0444           RSB
                          0445
                          0445  DEC_SHMCEB_REF:
          7E     51   7D  0445           MOVQ     R1,-(SP)                    ; SAVE REGISTERS
     51     15 A4     9A  0448           MOVZBL   SHB$B_PORT(R4),R1           ; GET PORT # FOR INDEX TO PROC REFCNT
     52     1D A6     9A  044C           MOVZBL   CEB$B_PROCCNT(R6),R2        ; GET # OF PROCESSOR SLAVE VA'S IN CEB
  52     52     02    78  0450           ASHL     #2,R2,R2                    ; CONVERT TO BYTE CNT, SLAVEVA=LONGWORD
         52     56    C0  0454           ADDL2    R6,R2                       ; OFFSET PAST THE SLAVE VA'S
              FBDC'   30  0457           BSBW     EXE$CEBREFLCK               ; ACQUIRE REF COUNT LOCK IN SHMCEB
           0D    50   E9  045A           BLBC     R0,10$                      ; BR IF UNABLE TO ACQUIRE LOCK
 38 A241     FFFF 8F  58  045D           ADAWI    #-1,CEB$L_VASLAVE1(R2)[R1] ; INCREMENT PROCESSOR REF COUNT
                 10   19  0464           BLSS     ERR_BUGCHK                  ; BUGCHECK IF ILLEGAL REFERENCE COUNT
        00 66    02   E7  0466           BBCCI    #CEB$V_REFCNTLCK,CEB$L_CEBFL(R6),10$ ; RELEASE REF CNT LOCK
           51    8E   7D  046A  10$:     MOVQ     (SP)+,R1                    ; RESTORE REGISTER
    01 0B A6     01   E1  046D           BBC      #CEB$V_PERM,CEB$B_STS(R6),20$ ; BR IF CLUSTER CAN BE DELETED
                      05  0472           RSB
              FB8A'   31  0473  20$:     BRW      EXE$SHMCEBDEL               ; TRY TO RELEASE CEB
                          0476
                          0476  ERR_BUGCHK:
                          0476           BUG_CHECK        CEBREFNEG,FATAL ; FATAL ERROR
```

```
                        047A            .SBTTL  EXE$SHMCEBDEL - TRY TO DELETE (RELEASE) MASTER CEB IN SH MEM
                        047A
                        047A    ;++
                        047A    ; FUNCTIONAL DESCRIPTION:
                        047A    ;
                        047A    ; THIS ROUTINE PERFORMS TWO FUNCTIONS:  (1) IF THE DELETE PENDING FLAG IS
                        047A    ; SET, THEN IT INFORMS THE OTHER PROCESSORS ON THE SHARED MEMORY THAT THE
                        047A    ; STATUS OF THE CEB IS NOW CHANGED, AND (2) IT CHECKS WHETHER OR NOT THE
                        047A    ; SHARED MEMORY CEB CAN BE RELEASED IMMEDIATELY.
                        047A    ;
                        047A    ; THE SHARED MEMORY CEB CAN ONLY BE RELEASED IF ALL THE REFERENCE COUNTS
                        047A    ; ARE ZERO (ONE COUNT PER PROCESSOR).  IF A COUNT IS NOT ZERO, THEN ONE OF THE
                        047A    ; FOLLOWING IS OUTSTANDING:
                        047A    ;       1) A SLAVE CEB STILL EXISTS FOR THE MASTER CEB
                        047A    ;       2) AN INTER-PROCESSOR REQUEST MESSAGE EXISTS FOR THE MASTER CEB
                        047A    ;       3) A PROCESSOR IS REFERENCING THE MASTER DURING A SHM CEB TBL SEARCH
                        047A    ;
                        047A    ; CALLING SEQUENCE:
                        047A    ;       JSB     EXE$SHMCEBDEL
                        047A    ;
                        047A    ; INPUT PARAMETERS:
                        047A    ;
                        047A    ;       R5 - SHARED MEMORY COMMON DATA PAGE ADDRESS
                        047A    ;       R6 - MASTER CEB ADDRESS
                        047A    ;
                        047A    ; IMPLICIT INPUTS:
                        047A    ;
                        047A    ;       NONE
                        047A    ;
                        047A    ; OUTPUT PARAMETERS:
                        047A    ;
                        047A    ;       R0 - STATUS CODE
                        047A    ;
                        047A    ; IMPLICIT OUTPUTS:
                        047A    ;
                        047A    ;       NONE
                        047A    ;
                        047A    ; COMPLETION CODES:
                        047A    ;
                        047A    ;       VARIOUS SYSTEM STATUS CODES.
                        047A    ;
                        047A    ; SIDE EFFECTS:
                        047A    ;
                        047A    ;       THE MASTER CEB MAY BE DELETED.  THE OTHER PROCESSORS MAY BE INFORMED
                        047A    ;       OF A CHANGE IN STATUS OF THE CEB.
                        047A    ;
                        047A    ;--
                        047A
                    00000000            .PSECT  AEXENONPAGED,BYTE
                        0000    EXE$SHMCEBDEL::
        7E  51      7D  0000            MOVQ    R1,-(SP)                ; SAVE REGISTER
   51   1D  A6      9A  0003    40$:    MOVZBL  CEB$B_PROCCNT(R6),R1    ; GET # OF REFERENCE COUNTS TO CHECK
52      38  A641    DE  0007            MOVAL   CEB$L_VASLAVE1(R6)[R1],R2 ; GET ADR OF 1ST REF COUNT
            0027    30  000C            BSBW    EXE$CEBREFLCK           ; ACQUIRE REF COUNT LOCK
        20  50      E9  000F            BLBC    R0,90$                  ; IGNORE POTENTIAL INTERLOCK ERROR
            82      B5  0012    50$:    TSTW    (R2)+                   ; ANY REFERENCES?
            18      12  0014            BNEQ    80$                     ; BR IF THERE IS AN OUTSTANDING REF
```

```
        F9 51  F5 0016  904         SOBGTR  R1,50$                              ; REPEAT FOR EACH REF CNT
     00 66  01  E6 0019  905         BBSSI   #CEB$V_LOCKED,CEB$L_CEBFL(R6),60$ ; LOCK IT AGAINST SEARCHES
     09 66  00  E7 001D  906 60$:    BBCCI   #CEB$V_VALID,CEB$L_CEBFL(R6),70$ ; MARK IT NO LONGER VALID
     51  1E A6  9A 0021  907         MOVZBL  CEB$B_CREATPORT(R6),R1   ; GET INDEX TO PORT QUOTA
   7C A541  01  58 0025  908         ADAWI   #1,SHD$W_CEFQUOTA(R5)[R1] ; RETURN PORT QUOTA
     00 66  01  E7 002A  909 70$:    BBCCI   #CEB$V_LOCKED,CEB$L_CEBFL(R6),80$ ; RELEASE GSD FOR OTHER USE
     00 66  02  E7 002E  910 80$:    BBCCI   #CEB$V_REFCNTLCK,CEB$L_CEBFL(R6),90$ ; RELEASE REF CNT LOCK
                   0032  911
        51  8E  7D 0032  912 90$:    MOVQ    (SP)+,R1                         ; RESTORE REGISTERS AND STATUS CODE
                05 0035  913         RSB                                      ; RETURN TO CALLER
```

```
                0036          .SBTTL  UNIQUE_CEB - CHECK THAT SH MEM CEB IS UNIQUE
                0036
                0036  ;++
                0036  ; FUNCTIONAL DESCRIPTION:
                0036  ;
                0036  ; THIS ROUTINE IS CALLED AFTER A SHARED MEMORY MASTER CEB IS INITIALIZED.
                0036  ; A SEARCH OF THE SPECIFIC SHARED MEMORY'S CEB TABLE IS MADE TO ASCERTAIN IF A
                0036  ; COMMON EVENT FLAG CLUSTER OF THE SAME NAME WAS CREATED DURING THE TIME THAT
                0036  ; $ASCEFC WAS CREATING THE SECTION.
                0036  ;
                0036  ; TWO LOCKS MUST BE ACQUIRED BEFORE THE SHARED MEMORY CEB TABLE MAY BE SEARCHED
                0036  ; TO VERIFY A SECTION IS UNIQUE.  THE FIRST IS THE SHARED MEMORY CEB MUTEX
                0036  ; WHICH INTERLOCKS PROCESSES ON ONE PROCESSOR.  THE SECOND IS THE SHARED MEMORY
                0036  ; CEB TABLE LOCK CONTAINED IN THE SHARED MEMORY COMMON DATA PAGE, WHICH
                0036  ; INTERLOCKS BETWEEN PROCESSORS.
                0036  ;
                0036  ; THE REGISTER CONVENTION (R6=ADR OF MASTER CEB JUST CREATED AND R11=ADR OF
                0036  ; MASTER CEB THAT IS THE DUPLICATE) IS SWITCHED DURING THIS ROUTINE TO MAKE
                0036  ; THE ROUTINES INC_SHMCEB_REF AND DEC_SHMCEB_REF LOCK THE DUPLICATE ENTRY.
                0036  ;
                0036  ; CALLING SEQUENCE:
                0036  ;
                0036  ;       BSBW    UNIQUE_CEB
                0036  ;
                0036  ; INPUT PARAMETERS:
                0036  ;
                0036  ;       R4 - ADDRESS OF SHARED MEMORY CONTROL BLOCK
                0036  ;       R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
                0036  ;       R6 - ADDRESS OF COMMON EVENT BLOCK TO BE VERIFIED AS UNIQUE
                0036  ;
                0036  ; IMPLICIT INPUTS:
                0036  ;
                0036  ;       NONE
                0036  ;
                0036  ; OUTPUT PARAMETERS:
                0036  ;
                0036  ;       R11 - 0 IF THE CEB IS UNIQUE
                0036  ;             OTHERWISE, ADDRESS OF DUPLICATE CEB
                0036  ;
                0036  ; IMPLICIT OUTPUTS:
                0036  ;
                0036  ;       NONE
                0036  ;
                0036  ; COMPLETION CODES:
                0036  ;
                0036  ;       NONE
                0036  ;
                0036  ; SIDE EFFECTS:
                0036  ;
                0036  ;       NONE
                0036  ;
                0036  ;--
                0036
            0000047A          .PSECT  Y$EXEPAGED,BYTE
                047A  UNIQUE_CEB:
                047A          .ENABL  LSB
034F 8F   BB    047A          PUSHR   #^M<R0,R1,R2,R3,R6,R8,R9> ; SAVE REGISTERS
```

```
           5B    56   D0  047E         MOVL    R6,R11                  ; REMEMBER ADR OF MASTER CEB JUST CREAT
               0055   30  0481         BSBW    SHMCEFTBLK              ; GET SHM MUTEX AND BIT LOCK
              4E 50   E9  0484         BLBC    R0,ERROR_EXIT           ; REPORT UNABLE TO GET BIT LOCK
   00A2 C5    15 A4   90  0487         MOVB    SHB$B_PORT(R4),SHD$B_CEFLOCK(R5) ; SET OWNER OF CEB TBL LOCK
      56 55   08 A5   C1  048D         ADDL3   SHD$L_CEFPTR(R5),R5,R6  ; GET ADR OF FIRST CEB IN SH MEM
         58   08 A6   3C  0492         MOVZWL  CEB$W_SIZE(R6),R8       ; GET SIZE OF ONE SHMCEB
         59   1C A5   3C  0496         MOVZWL  SHD$W_CEFMAX(R5),R9     ; GET # OF CEB'S IN SHM TBL
               FF83   30  049A  30$:   BSBW    INC_SHMCEB_REF          ; LOCK CEB AGAINST DELETION
              1F 50   E9  049D         BLBC    R0,45$                  ; DONT USE ENTRY IF IT IS REFCNT LOCKED
      18 66     01    E0  04A0         BBS     #CEB$V_LOCKED,CEB$L_CEBFL(R6),40$ ;BR IF CEB LOCKED FOR READING
      14 66     00    E1  04A4         BBC     #CEB$V_VALID,CEB$L_CEBFL(R6),40$ ;BR IF CEB IS NOT IN USE
   22 AB    22 A6     B1  04A8         CMPW    CEB$W_GRP(R6),CEB$W_GRP(R11) ; ARE CEB'S IN SAME GROUP?
                0D    12  04AD         BNEQ    40$                     ; BR IF NOT, KEEP LOOKING
28 A6    28 AB  10    29  04AF         CMPC3   #16,CEB$T_EFCNAM(R11),CEB$T_EFCNAM(R6) ; CEB'S HAVE SAME NAME?
                05    12  04B5         BNEQ    40$                     ; BR IF DIFFERENT NAMES
           5B   56    D1  04B7         CMPL    R6,R11                  ; IS THIS THE CEB JUST MADE?
                0B    12  04BA         BNEQ    50$                     ; BR IF DIFFERENT, FOUND DUPLICATE
              FF86    30  04BC  40$:   BSBW    DEC_SHMCEB_REF          ; RELEASE LOCK AGAINST DELETION
                          04BF                                         ; IGNORE POTENTIAL INTERLOCK ERROR
           56   58    C0  04BF  45$:   ADDL2   R8,R6                   ; GET ADR OF NEXT SHMCEB
              D5 59   F5  04C2         SOBGTR  R9,30$                  ; LOOK AT EACH ENTRY IN SHM TBL
                56    D4  04C5         CLRL    R6                      ; INDICATE NO DUPLICATE ENTRY
   00 009F C5   04    E7  04C7  50$:   BBCCI   #SHD$V_CEFLCK,SHD$B_FLAGS(R5),60$ ; RELEASE SHM CEB TBL LOCK
           5B   56    D0  04CD  60$:   MOVL    R6,R11                  ; RETURN DUPLICATE ENTRY ADR IN R11
         034F   8F    BA  04D0         POPR    #^M<R0,R1,R2,R3,R6,R8,R9> ; RESTORE REGISTERS
                05        04D4         RSB                             ; RETURN TO $ASCEFC
                          04D5
                          04D5  ; ************
                          04D5  ; AT SOME LATER DATE, THIS SHOULD SEND AN ERROR MESSAGE TO THE ERROR LOGGER.
                          04D5  ; ************
                          04D5  ERROR_EXIT:
                56    D4  04D5         CLRL    R6                      ; FAILURE TO ACQUIRE BIT LOCK
                F4    11  04D7         BRB     60$                     ; RETURN ERROR STATUS
                          04D9         .DSABL  LSB
```

```
                          04D9        .SBTTL EXE$CEBREFLCK - ACQUIRE SHMCEB REFERENCE COUNT LOCK
                          04D9
                          04D9  ;++
                          04D9  ; FUNCTIONAL DESCRIPTION:
                          04D9  ;
                          04D9  ; THE ROUTINE IS CALLED TO ACQUIRE EXCLUSIVE USE OF A SHARED MEMORY COMMON
                          04D9  ; EVENT FLAG BLOCK.  THIS IS DONE BY LOOPING TO ACQUIRE A BIT LOCK IN THE
                          04D9  ; MASTER CEB.  IF THE BIT LOCK COULD NOT BE ACQUIRED, THEN AN ERROR CODE IS
                          04D9  ; RETURNED.
                          04D9  ;
                          04D9  ;
                          04D9  ; CALLING SEQUENCE:
                          04D9  ;
                          04D9  ;       BSBW    EXE$CEBREFLCK
                          04D9  ;
                          04D9  ; INPUT PARAMETERS:
                          04D9  ;
                          04D9  ;       R6 - ADDRESS OF SHARED MEMORY COMMON EVENT BLOCK
                          04D9  ;
                          04D9  ; IMPLICIT INPUTS:
                          04D9  ;
                          04D9  ;       NONE
                          04D9  ;
                          04D9  ; OUTPUT PARAMETERS:
                          04D9  ;
                          04D9  ;       R0 - STATUS CODE
                          04D9  ;
                          04D9  ; IMPLICIT OUTPUTS:
                          04D9  ;
                          04D9  ;       NONE
                          04D9  ;
                          04D9  ; COMPLETION CODES:
                          04D9  ;
                          04D9  ;       SS$_NORMAL - SUCCESSFULLY ACQUIRED LOCK
                          04D9  ;       SS$_INTERLOCK  - UNABLE TO ACQUIRE LOCK
                          04D9  ;
                          04D9  ; SIDE EFFECTS:
                          04D9  ;
                          04D9  ;       NONE
                          04D9  ;
                          04D9  ;--
                          04D9
                      00000036        .PSECT  AEXENONPAGED,BYTE
                          0036  EXE$CEBREFLCK::
                51    DD  0036        PUSHL   R1                          ; SAVE REGISTER
51       00000000'GF    D0  0038        MOVL    G^EXE$GL_LOCKRTRY,R1         ; GET LOOP COUNT FOR BIT LOCK
      07 66   02    E6  003F  10$:    BBSSI   #CEB$V_REFCNTLCK,CEB$L_CEBFL(R6),20$ ; TRY TO ACQUIRE BIT LOCK
         50   01    9A  0043        MOVZBL  #SS$_NORMAL,R0               ; REPORT LOCK SUCCESSFULLY ACQUIR
            51 8ED0     0046        POPL    R1                          ; RESTORE REGISTER
                  05    0049        RSB                                 ; RETURN SUCCESS CODE
         F2 51    F5    004A  20$:    SOBGTR  R1,10$                      ; TRY AGAIN TO ACQUIRE BIT LOCK
            51 8ED0     004D        POPL    R1                          ; RESTORE REGISTER
50       038C 8F  3C    0050        MOVZWL  #SS$_INTERLOCK,R0            ; REPORT ERROR STATUS
                  05    0055        RSB                                 ; RETURN TO CALLER
```

```
                     0056        .SBTTL  SHMCEFTBLK - ACQUIRE EXCLUSIVE USE OF SHARED MEMORY TABLE
                     0056
                     0056  ;++
                     0056  ; FUNCTIONAL DESCRIPTION:
                     0056  ;
                     0056  ; THE ROUTINE IS CALLED TO ACQUIRE EXCLUSIVE USE OF A SHARED MEMORY COMMON
                     0056  ; EVENT FLAG TABLE.  THIS IS DONE BY LOOPING TO ACQUIRE A SHARED MEMORY BIT
                     0056  ; LOCK.  IF THE BIT LOCK COULD NOT BE ACQUIRED, THEN AN ERROR CODE IS RETURNED.
                     0056  ;
                     0056  ; CALLING SEQUENCE:
                     0056  ;
                     0056  ;       BSBW    SHMCEFTBLK
                     0056  ;
                     0056  ; INPUT PARAMETERS:
                     0056  ;
                     0056  ;       R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
                     0056  ;
                     0056  ; IMPLICIT INPUTS:
                     0056  ;
                     0056  ;       NONE
                     0056  ;
                     0056  ; OUTPUT PARAMETERS:
                     0056  ;
                     0056  ;       R0 - STATUS CODE
                     0056  ;
                     0056  ; IMPLICIT OUTPUTS:
                     0056  ;
                     0056  ;       NONE
                     0056  ;
                     0056  ; COMPLETION CODES:
                     0056  ;
                     0056  ;       SS$_NORMAL - SUCCESSFULLY ACQUIRED LOCK
                     0056  ;       SS$_INTERLOCK  - UNABLE TO ACQUIRE LOCK
                     0056  ;
                     0056  ; SIDE EFFECTS:
                     0056  ;
                     0056  ;       NONE
                     0056  ;
                     0056  ;--
                     0056
                 000004D9        .PSECT  Y$EXEPAGED,BYTE
                     04D9  SHMCEFTBLK:
             51   DD 04D9        PUSHL   R1                          ; SAVE REGISTER
51  00000000'GF   D0 04DB        MOVL    G^EXE$GL_LOCKRTRY,R1        ; GET LOOP COUNT FOR BIT LOCK
07 009F C5   04   E6 04E2  10$:  BBSSI   #SHD$V_CEFLCK,SHD$B_FLAGS(R5),20$ ; TRY TO ACQUIRE BIT LOCK
        50   01   9A 04E8        MOVZBL  #SS$_NORMAL,R0              ; REPORT LOCK SUCCESSFULLY ACQUIR
             51 8ED0 04EB        POPL    R1                          ; RESTORE REGISTER
                  05 04EE        RSB                                 ; RETURN SUCCESS CODE
          F0 51   F5 04EF  20$:  SOBGTR  R1,10$                      ; TRY AGAIN TO ACQUIRE BIT LOCK
             51 8ED0 04F2        POPL    R1                          ; RESTORE REGISTER
  50    038C 8F   3C 04F5        MOVZWL  #SS$_INTERLOCK,R0           ; REPORT ERROR STATUS
                  05 04FA        RSB                                 ; RETURN TO CALLER
```

```
                          04FB          .SBTTL  SHD_FIND_SHB - FIND SH MEM CONTROL BLOCK FOR SPECIFIC SHD
                          04FB
                          04FB  ;++
                          04FB  ; FUNCTIONAL DESCRIPTION:
                          04FB  ;
                          04FB  ; THIS ROUTINE TAKES COMMON DATA PAGE ADDRESS AND FINDS THE SHARED MEMORY
                          04FB  ; CONTROL BLOCK ADDRESS FOR THIS SHARED MEMORY.
                          04FB  ;
                          04FB  ; CALLING SEQUENCE:
                          04FB  ;
                          04FB  ;       BSBW    SHD_FIND_SHB
                          04FB  ;
                          04FB  ; INPUT PARAMETERS:
                          04FB  ;
                          04FB  ;       R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
                          04FB  ;
                          04FB  ; IMPLICIT INPUTS:
                          04FB  ;
                          04FB  ;       NONE
                          04FB  ;
                          04FB  ; OUTPUT PARAMETERS:
                          04FB  ;
                          04FB  ;       R0 - STATUS CODE
                          04FB  ;       R4 - SHB ADR, OR 0
                          04FB  ;
                          04FB  ; IMPLICIT OUTPUTS:
                          04FB  ;
                          04FB  ;       NONE
                          04FB  ;
                          04FB  ; COMPLETION CODES:
                          04FB  ;
                          04FB  ;       SS$_NORMAL - SUCCESSFULLY FOUND SHB ADR
                          04FB  ;       SS$_SHMNOTCNCT - SHARED MEMORY NOT CONNECTED
                          04FB  ;
                          04FB  ; SIDE EFFECTS:
                          04FB  ;
                          04FB  ;       NONE
                          04FB  ;
                          04FB  ;--
                          04FB
                      000004FB          .PSECT  Y$EXEPAGED,BYTE
                          04FB  SHD_FIND_SHB:
                          04FB          .ENABL  LSB
           50    01    9A 04FB          MOVZBL  #SS$_NORMAL,R0          ; ASSUME SUCCESS
   54  00000000'EF    D0 04FE          MOVL    EXE$GL_SHBLIST,R4       ; GET ADR OF FIRST SHB
                 11    13 0505  10$:    BEQL    NOT_FOUND               ; BR IF NO SHB ADR
     07 0B A4    00    E1 0507          BBC     #SHB$V_CONNECT,SHB$B_FLAGS(R4),GET_NXT_SHM ; BR IF MEM NOT CONCT
        55    04 A4    D1 050C          CMPL    SHB$L_DATAPAGE(R4),R5   ; DOES THE COM DATA PAGE ADR MATCH?
                 01    12 0510          BNEQ    GET_NXT_SHM             ; BR IF NO MATCH, FIND NEXT SHB ADR
                       05 0512          RSB                             ; SHB FOUND, RETURN ADR
                          0513  GET_NXT_SHM:
           54    64    D0 0513          MOVL    SHB$L_LINK(R4),R4       ; GET NEXT SHB IN LIST
                 ED    11 0516          BRB     10$                     ; CONTINUE
                          0518  NOT_FOUND:
     50   037C 8F    3C 0518          MOVZWL  #SS$_SHMNOTCNCT,R0      ; NO MORE MEM, REPORT SHB NOT FOUND
                       05 051D          RSB
                          051E          .DSABL  LSB
```

```
 051E  1172
 051E  1173
 051E  1174          .END
```

```
ALOSHMCEB                         00000398 R      03
ASSOCIATE                         00000188 R      03
BUG$_CEBREFNEG                    ********    X   03
CEB$B_CREATPORT                 = 0000001E
CEB$B_DELETPORT                 = 0000001F
CEB$B_LOCK                      = 0000001C
CEB$B_PROCCNT                   = 0000001D
CEB$B_STS                       = 0000000B
CEB$B_TYPE                      = 0000000A
CEB$L_CEBFL                     = 00000000
CEB$L_EFC                       = 00000010
CEB$L_MASTER                    = 00000040
CEB$L_PID                       = 0000000C
CEB$L_SHB                       = 00000038
CEB$L_UIC                       = 00000020
CEB$L_VASLAVE1                  = 00000038
CEB$L_WQBL                      = 00000018
CEB$L_WQFL                      = 00000014
CEB$T_EFCNAM                    = 00000028
CEB$V_LOCKED                    = 00000001
CEB$V_NOQUOTA                   = 00000000
CEB$V_PERM                      = 00000001
CEB$V_REFCNTLCK                 = 00000002
CEB$V_VALID                     = 00000000
CEB$W_GRP                       = 00000022
CEB$W_INDX                      = 0000003C
CEB$W_PROT                      = 00000024
CEB$W_REFC                      = 00000026
CEB$W_SIZE                      = 00000008
CEB$W_STATE                     = 0000001E
CEB$W_WQCNT                     = 0000001C
CEBLOCK                           0000027C R      03
DEC_SHMCEB_REF                    00000445 R      03
DISASOC                           00000194 R      03
DYN$C_SLAVCEB                   = 0000002D
EFN                             = 00000004
ERROR_EXIT                        000004D5 R      03
ERR_BUGCHK                        00000476 R      03
EXE$ALLOCCEB                      ********    X   03
EXE$ASCEFC                        00000027 RG     03
EXE$CEBREFLCK                     00000036 RG     04
EXE$DACEFC                        00000000 RG     03
EXE$DEANONPAGED                   ********    X   03
EXE$DLCEFC                        00000215 RG     03
EXE$GL_CEBMTX                     ********    X   03
EXE$GL_LOCKRTRY                   ********    X   04
EXE$GL_SHBLIST                    ********    X   03
EXE$SHMCEBDEL                     00000000 RG     04
EXIT                              000001FD R      03
EXITE                             000001FF R      03
EXITN                             000001FA R      03
FIND_SHM_CEB                      0000032A R      03
GETCLUSTER                        00000015 R      03
GET_NXT_SHM                       00000513 R      03
INC_SHMCEB_REF                    00000420 R      03
IPL$_SYNCH                      = 00000008
JIB$C_MPID                      = 00000054
JIB$W_TQCNT                     = 00000034
LOCLUST                           00000289 R      03
MMG$CEFTRNLOG                     ********    X   03
MMG$FINDSHB                       ********    X   03
NAME                            = 00000008
NOT_FOUND                         00000518 R      03
NO_FREE_CEB                       00000407 R      03
NO_QUOTA                          0000040E R      03
PCB$L_EFC2P                     = 00000058
PCB$L_JIB                       = 00000080
PCB$L_PHD                       = 0000006C
PCB$L_PID                       = 00000060
PCB$L_UIC                       = 000000BC
PCB$W_GRP                       = 000000BE
PERM                            = 00000010
PR$_IPL                         = 00000012
PROT                            = 0000000C
PRV$V_PRMCEB                    = 0000000A
PRV$V_SHMEM                     = 0000001B
RETQUOTA                          00000000 R      01
SCH$C_CEF                       = 00000003
SCH$GL_PCBVEC                     ********    X   01
SCH$GQ_CEBHD                      ********    X   03
SCH$GW_CEBCNT                     ********    X   03
SCH$LOCKW                         ********    X   03
SCH$UNLOCK                        ********    X   03
SHB$B_FLAGS                     = 0000000B
SHB$B_PORT                      = 00000015
SHB$L_DATAPAGE                  = 00000004
SHB$L_LINK                      = 00000000
SHB$V_CONNECT                   = 00000000
SHD$B_CEFLOCK                   = 000000A2
SHD$B_FLAGS                     = 0000009F
SHD$L_CEFPTR                    = 00000008
SHD$V_CEFLCK                    = 00000004
SHD$W_CEFMAX                    = 0000001C
SHD$W_CEFQUOTA                  = 0000007C
SHD_FIND_SHB                      000004FB R      03
SHMCEFTBLCK                       000004D9 R      03
SS$_ACCVIO                      = 0000000C
SS$_EXPORTQUOTA                 = 000003AC
SS$_EXQUOTA                     = 0000001C
SS$_ILLEFC                      = 000000EC
SS$_INTERLOCK                   = 0000038C
SS$_IVLOGNAM                    = 00000154
SS$_NOPRIV                      = 00000024
SS$_NORMAL                      = 00000001
SS$_NOSHMBLOCK                  = 000003B4
SS$_SHMNOTCNCT                  = 0000037C
UNIQUE_CEB                        0000047A R      03
```

```
                                   +----------------+
                                   ! Psect synopsis !
                                   +----------------+

PSECT name                Allocation          PSECT No.  Attributes
----------                ----------          ---------  ----------
.  ABS  .                 00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
. BLANK .                 0000001F  (   31.)  01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$ABS$                     00000000  (    0.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
Y$EXEPAGED                0000051E  ( 1310.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
AEXENONPAGED              00000056  (   86.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                    36    00:00:00.05    00:00:01.22
Command processing               125    00:00:00.59    00:00:03.35
Pass 1                           356    00:00:11.89    00:00:34.27
Symbol table sort                  0    00:00:01.65    00:00:05.37
Pass 2                           221    00:00:03.29    00:00:11.87
Symbol table output               14    00:00:00.10    00:00:00.41
Psect synopsis output              2    00:00:00.03    00:00:00.03
Cross-reference output             0    00:00:00.00    00:00:00.00
Assembler run totals             756    00:00:17.61    00:00:56.53

The working set limit was 1650 pages.
71103 bytes (139 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1081 non-local and 74 local symbols.
1174 source lines were read in Pass 1, producing 19 object records in Pass 2.
25 pages of virtual memory were used to define 24 macros.

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

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          14
_$255$DUA28:[SYSLIB]STARLET.MLB;2                        7
TOTALS (all libraries)                                  21

1184 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSASCEFC/OBJ=OBJ$:SYSASCEFC MSRC$:SYSASCEFC/UPDATE=(ENH$:SYSASCEFC)+EXECML$/LIB
```

0381 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SSDEF
LIS

SYSASCEFC
LIS

SYSACPFDT
LIS

SYSADJSTK
LIS

SYSBRKTHR
LIS

SYSASSIGN
LIS

SWAPPER
LIS

SYSASTCON
LIS

SVAPTE
LIS

SYSADJWSL
LIS