```
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDD         DDD  CCC              LLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
```

```
**FILE**ID**LOGICAL                                         N 14



LL            OOOOOO      GGGGGGGG    IIIIII      CCCCCCCC    AAAAAA    LL
LL            OOOOOO      GGGGGGGG    IIIIII      CCCCCCCC    AAAAAA    LL
LL          OO      OO  GG              II      CC          AA      AA  LL
LL          OO      OO  GG              II      CC          AA      AA  LL
LL          OO      OO  GG              II      CC          AA      AA  LL
LL          OO      OO  GG              II      CC          AA      AA  LL
LL          OO      OO  GG              II      CC          AA      AA  LL
LL          OO      OO  GG              II      CC          AA      AA  LL
LL          OO      OO  GG  GGGGGG      II      CC          AAAAAAAAAA  LL
LL          OO      OO  GG  GGGGGG      II      CC          AAAAAAAAAA  LL
LL          OO      OO  GG      GG      II      CC          AA      AA  LL              ....
LL          OO      OO  GG      GG      II      CC          AA      AA  LL              ....
LLLLLLLLLL    OOOOOO      GGGGGG      IIIIII      CCCCCCCC  AA      AA  LLLLLLLLLL      ....
LLLLLLLLLL    OOOOOO      GGGGGG      IIIIII      CCCCCCCC  AA      AA  LLLLLLLLLL      ....

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

Table of contents

```
    (3)      130      ALLOCATE DEVICE
    (4)      271      ASSIGN LOGICAL NAME TO EQUIVALENCE STRING
    (5)      357      DEFINE LOGICAL NAME EQUIVALENCE
    (7)      482      DEALLOCATE DEVICE
    (8)      515      DEASSIGN LOGICAL NAME EQUIVALENCE
    (9)      563      TEST IF LOGICAL NAME IS SYS$OUTPUT
    (10)     591      PROCESS COMMON COMMAND QUALIFIERS
    (11)     740      GET TRANSLATION ATTRIBUTES
    (12)     772      CREATE LOGICAL NAME TABLE
    (13)     945      SHOW LOGICAL NAME EQUIVALENCES
```

```
0000          .TITLE  LOGICAL - LOGICAL NAME COMMANDS
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    ; LOGICAL NAME DCLS COMMAND EXECUTION
0000    ;
0000    ;       ALLOCATE DEVICE
0000    ;       ASSIGN LOGICAL NAME
0000    ;       DEALLOCATE DEVICE
0000    ;       DEASSIGN LOGICAL NAME
0000    ;       DEFINE LOGICAL NAME
0000    ;       CREATE LOGICAL NAME TABLE
0000    ;       SHOW LOGICAL NAME TRANSLATION
0000    ;
0000    ; Peter George 20-April-1983
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-007 HWS0078         Harold Schultz          02-Jul-1984
0000    ;               Fix negation of table qualifiers in ASSIGN, DEASSIGN,
0000    ;               and DEFINE commands.
0000    ;
0000    ;       V03-006 HWS0041         Harold Schultz          12-Apr-1984
0000    ;               Add ALLOCATE /GENERIC.
0000    ;
0000    ;       V03-005 PCG0003         Peter George            20-Mar-1984
0000    ;               Add /JOB qualifier.
0000    ;
0000    ;       V03-004 HWS0005         Harold Schultz          07-Feb-1984
0000    ;               Added /PROTECTION=(SY:RWED,OW:RWED,...) qualifier for
0000    ;               when creating a logical name table.
0000    ;               Add /LOG qualifier when creating a logical name table.
0000    ;               Output informational messages after table creation.
0000    ;
```

```
0000    ;       V03-003 TMK0001         Todd M. Katz            12-Oct-1983
0000    ;               Translate logical names using LNM$DCL_LOGICAL as the table
0000    ;               name instead of LNM$DEFAULT_SEARCH.
0000    ;
0000    ;       V03-002 PCG0002         Peter George            01-Jul-1983
0000    ;               Fix bug in ALLOCATE command parsing.
0000    ;               Replace old logical name commands.
0000    ;               Stop fooling around with the CRELOG bit.
0000    ;
0000    ;       V03-001 PCG0001         Peter George            15-Jun-1983
0000    ;               Return more helpful status when ALLOCATE fails.
0000    ;               Pass equivalence name to DCL$OPEN_OUTPUT.
0000    ;---
```

```
                                    0000       ;
                                    0000       ; MACRO LIBRARY CALLS
                                    0000       ;
                                    0000
                                    0000               PRCDEF                          ;DEFINE PROCESS WORK AREA
                                    0000               WRKDEF                          ;DEFINE COMMAND WORK AREA
                                    0000               PTRDEF                          ;DEFINE RESULT PARSE DESCRIPTOR FORMAT
                                    0000               $CLIMSGDEF                      ;DEFINE ERROR/STATUS VALUES
                                    0000               $LNMDEF                         ;DEFINE LOGICAL NAME OFFSETS
                                    0000               $PSLDEF                         ;DEFINE PROCESSOR STATUS FIELDS
                                    0000               $SSDEF                          ;DEFINE SYSTEM STATUS VALUES
                                    0000
                                00000000               .PSECT  DCL$ZCODE,BYTE,RD,NOWRT
                                    0000
                                    0000       ;
                                    0000       ; LOCAL DATA
                                    0000       ;
                                    0000       OUTPUTNAM:
   54 55 50 54 55 4F 24 53 59 53 00' 0000               .ASCIC  'SYS$OUTPUT'
                                 0A  0000
                                    000B       LNM$PROCESS:
53 53 45 43 4F 52 50 24 4D 4E 4C 00' 000B               .ASCIC  'LNM$PROCESS'
                                 0B  000B
                                    0017       LNM$JOB:
            42 4F 4A 24 4D 4E 4C 00' 0017               .ASCIC  'LNM$JOB'
                                 07  0017
                                    001F       LNM$GROUP:
      50 55 4F 52 47 24 4D 4E 4C 00' 001F               .ASCIC  'LNM$GROUP'
                                 09  001F
                                    0029       LNM$SYSTEM:
   4D 45 54 53 59 53 24 4D 4E 4C 00' 0029               .ASCIC  'LNM$SYSTEM'
                                 0A  0029
                                    0034       LNM$DCL_LOGICAL:
47 4F 4C 5F 4C 43 44 24 4D 4E 4C 00' 0034               .ASCIC  'LNM$DCL_LOGICAL'
                        4C 41 43 49  0040
                                 0F  0034
                                    0044       LNM$PROCESS_DIRECTORY:
53 53 45 43 4F 52 50 24 4D 4E 4C 00' 0044               .ASCIC  'LNM$PROCESS_DIRECTORY'
      59 52 4F 54 43 45 52 49 44 5F  0050
                                 15  0044
                                    005A       LNM$FILE_DEV:
45 44 5F 45 4C 49 46 24 4D 4E 4C 00' 005A               .ASCIC  'LNM$FILE_DEV'
                                 56  0066
                                 0C  005A
                                    0067       UNDEFINED:
      44 45 4E 49 46 45 44 4E 55 00' 0067               .ASCIC  'UNDEFINED'
                                 09  0067
                                    0071       LOGICALMSG:
41 21 22 20 3D 20 53 41 21 20 20 00' 0071               .ASCIC  '  !AS = "!AS"  (!AS)'
         29 53 41 21 28 20 20 22 53  007D
                                 14  0071
                        52 57 45 44  0086       ACCESS: .ASCII  /DEWR/                  ;ACCESS PROTECTION CODES
                        53 4F 47 57  008A       CLASS:  .ASCII  /WGOS/                  ;PROTECTION CLASSES
                                    008E       ;
                                    008E       ; DEFINE OFFSETS FOR COMMON PARSING DATA STRUCTURE
                                    008E       ;
                       00000020     008E               EQUNAM = 32
```

```
00000018  008E   115        LOGNAM = 24
00000010  008E   116        TABNAM = 16
0000000C  008E   117        ACMODE = 12
00000008  008E   118        QUAL = 8
0000FF00  008E   119        DEF_PROT = ^XFF00
00000000  008E   120        LOG_V = 0
00000001  008E   121        LOG_M = 1
00000001  008E   122        ATTR_V = 1
00000002  008E   123        ATTR_M = 2
00000002  008E   124        DEF_V = 2
00000004  008E   125        DEF_M = 4
00000004  008E   126        NAME_ATTR = 4
00000000  008E   127        TRAN_ATTR = 0
          008E   128
```

```
                    008E           .SBTTL  ALLOCATE DEVICE
                    008E    ;+
                    008E    ; DCL$ALLOCATE - ALLOCATE DEVICE
                    008E    ;
                    008E    ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE ALLOCATE
                    008E    ; COMMAND.
                    008E    ;
                    008E    ; INPUTS:
                    008E    ;
                    008E    ;      R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                    008E    ;      R9 = ADDRESS OF SCRATCH STACK.
                    008E    ;      R10 = BASE ADDRESS OF COMMAND WORK AREA.
                    008E    ;      R11 = BASE ADDRESS OF PROCESS WORK AREA.
                    008E    ;
                    008E    ; OUTPUTS:
                    008E    ;
                    008E    ;      THE SPECIFIED DEVICE IS ALLOCATED AND ASSIGNED THE SPECIFIED LOGICAL
                    008E    ;      NAME. IF THE LOGICAL NAME WAS PREVIOUSLY ASSIGNED, THEN A MESSAGE TO
                    008E    ;      THAT EFFECT IS WRITTEN TO THE OUTPUT STREAM.
                    008E    ;-
                    008E    DCL$ALLOCATE::                          ;ALLOCATE DEVICE
                    008E    ;
                    008E    ; Allocate and init common logical name data structure.
                    008E    ;
      59    10  C2  008E           SUBL    #16,R9                   ;ALLOCATE PHYSICAL DEV BUFFER
      79    59  D0  0091           MOVL    R9,-(R9)                 ;INIT THE EQUIV NAME DESCR
      79    10  3C  0094           MOVZWL  #16,-(R9)                ;
            79  7C  0097           CLRQ    -(R9)                    ;ALLOCATE LOG NAME DESCR
  55   BE   AF  9E  0099           MOVAB   LNM$FILE_DEV,R5          ;SET LOGICAL NAME TABLE
      54    85  9A  009D           MOVZBL  (R5)+,R4                 ;
      79    54  7D  00A0           MOVQ    R4,-(R9)                 ;SAVE THE DESCRIPTOR
      79    02  D0  00A3           MOVL    #PSL$C_SUPER,-(R9)       ;AND SUPERVISOR MODE
      79    01  D0  00A6           MOVL    #LOG_M,-(R9)             ;ASSUME /LOG DEFAULTED
            79  7C  00A9           CLRQ    -(R9)                    ;SET DEFAULT NAME/TRAN ATTRIBUTES
      58    59  D0  00AB           MOVL    R9,R8                    ;COPY THE BASE OF THE DATA STRUCTURE
            79  D4  00AE           CLRL    -(R9)                    ;ALLOCATE /TYPE VALUE
                    00B0
            56  D4  00B0           CLRL    R6                       ;SET NO TERMINATOR YET
57    0908  8F  3C  00B2           MOVZWL  #SS$_NOSUCHDEV,R7        ;PRESET ERROR STATUS
                    00B7
                    00B7    ;
                    00B7    ; Process /log and /type command qualif-ers.
                    00B7    ;
         FF46'  30  00B7    10$:   BSBW    DCL$GETDVAL              ;GET NEXT TOKEN
      03    55  91  00BA           CMPB    R5,#PTR_K_PARAMETR       ;PARAMETER VALUE?
            2C  13  00BD           BEQL    40$                      ;YES, THEN NO /GENE OR /LOG
         FF3E'  30  00BF           BSBW    DCL$GETNVAL              ;GET QUALIFIER TYPE
  51    00'8F   91  00C2           CMPB    #CLI$K_ALLO_GENE,R1      ;IS IT /GENE
            0D  13  00C6           BEQL    20$                      ;YES, THEN BRANCH
  08 A8     01  C8  00C8           BISL    #LOG_M,QUAL(R8)          ;ASSUME /LOG
       E8   53  E9  00CC           BLBC    R3,10$                   ;BRANCH IF SO
  08 A8     01  CA  00CF           BICL    #LOG_M,QUAL(R8)          ;SET /NOLOG
            E2  11  00D3           BRB     10$                      ;GET NEXT TOKEN
            69  D4  00D5    20$:   CLRL    (R9)                     ;ASSUME /NOGENERIC
       DD   53  E8  00D7           BLBS    R3,10$                   ;BRANCH IF SO
      69    01  D0  00DA           MOVL    #1,(R9)                  ;IT WAS /GENERIC
            D8  11  00DD           BRB     10$                      ;GET NEXT TOKEN
```

```
                          00DF   187
                          00DF   188 ;
                          00DF   189 ; Return allocation error.
                          00DF   190 ;
         50   57    D0    00DF   191 90$:     MOVL    R7,R0                              ;SET ERROR STATUS
                    05    00E2   192          RSB                                        ;EXIT
                          00E3   193
                          00E3   194 ;
                          00E3   195 ; Process the device names.
                          00E3   196 ;
            FF1A'   30    00E3   197 30$:     BSBW    DCL$GETDVAL                        ;GET NEXT TOKEN
         05   56    91    00E6   198          CMPB    R6,#PTR_K_COMMA                    ;ANOTHER PARAMETER VALUE IN LIST?
              F4    12    00E9   199          BNEQ    90$                                ;NO, THEN ERROR
         03   55    91    00EB   200 40$:     CMPB    R5,#PTR_K_PARAMETR                 ;PARAMETER VALUE?
              EF    12    00EE   201          BNEQ    90$                                ;NO, THEN ERROR
         56   54    D0    00F0   202          MOVL    R4,R6                              ;SAVE TOKEN TERMINATOR
            FF0A'   30    00F3   203          BSBW    DCL$COMPSTRING                     ;REMOVE ANY DOUBLE QUOTES
       18 A8   51   7D    00F6   204          MOVQ    R1,LOGNAM(R8)                      ;SAVE THE DESCRIPTOR
    FF A241   3A    91    00FA   205          CMPB    #^A/:/,-1(R2)[R1]                  ;DEVICE NAME END WITH A COLON?
              03    12    00FF   206          BNEQ    50$                                ;IF NEQ NO
           18 A8    D7    0101   207          DECL    LOGNAM(R8)                         ;REDUCE LENGTH OF DEVICE NAME
         50   69    D0    0104   208 50$:     MOVL    (R9),R0                            ;GET /GENERIC INDICATOR
                          0107   209          $ALLOC_S  LOGNAM(R8),EQUNAM(R8),-          ;ALLOCATE DEVICE
                          0107   210                  EQUNAM(R8),#0,R0                   ;
         57   50    D0    011B   211          MOVL    R0,R7                              ;SAVE FINAL STATUS
         C2 50      E9    011E   212          BLBC    R0,30$                             ;IF ERROR, TRY NEXT DEVICE IN LIST
                          0121   213
                          0121   214 ;
                          0121   215 ; Output the device allocated message.
                          0121   216 ;
   10 08 A8   00    E1    0121   217          BBC     #LOG_V_QUAL(R8),80$                ;SKIP IF /NOLOG
           20 A8    7F    0126   218          PUSHAQ  EQUNAM(R8)                         ;PUSH DESCRIPTOR ADDRESS
         51   01    D0    0129   219          MOVL    #1,R1                              ;SET ARG COUNT
 50   0003DDE3 8F   D0    012C   220          MOVL    #CLI$_ALLOC,R0                     ;SET STATUS
            FECA'   30    0133   221          BSBW    DCL$FORMMSG                        ;OUTPUT INFORMATIONAL MESSAGE
                          0136   222
                          0136   223 ;
                          0136   224 ; Get the requested logical name.
                          0136   225 ;
            FEC7'   30    0136   226 80$:     BSBW    DCL$GETDVAL                        ;GET NEXT PARAMETER VALUE
         03   55    91    0139   227          CMPB    R5,#PTR_K_PARAMETR                 ;PARAMETER VALUE?
              02    13    013C   228          BEQL    60$                                ;CONTINUE IF LOGICAL NAME FOUND
              5B    11    013E   229          BRB     95$                                ;EXIT IF NOT
         05   56    91    0140   230 60$:     CMPB    R6,#PTR_K_COMMA                    ;STILL IN P1 LIST?
              05    12    0143   231          BNEQ    70$                                ;IF P2 FOUND, ASSIGN THE LOGICAL NAM
         56   54    D0    0145   232          MOVL    R4,R6                              ;COPY TERMINATOR TYPE CODE
              EC    11    0148   233          BRB     80$                                ;LOOP UNTIL P2 OR EOL FOUND
                          014A   234
            FEB3'   30    014A   235 70$:     BSBW    DCL$COMPSTRING                     ;REMOVE QUOTATION MARKS
    FF A241   3A    91    014D   236          CMPB    #^A/:/,-1(R2)[R1]                  ;LOGICAL NAME END WITH COLON?
              02    12    0152   237          BNEQ    75$                                ;IF NEQ NO
              51    D7    0154   238          DECL    R1                                 ;REDUCE LENGTH OF LOGICAL NAME
       18 A8   51   7D    0156   239 75$:     MOVQ    R1,LOGNAM(R8)                      ;SAVE THE LOGICAL NAME
                          015A   240
                          015A   241 ;
                          015A   242 ; Create the required item list.
                          015A   243 ;
```

```
                  7E    7C  015A          CLRQ     -(SP)                            ;TERMINATE THE LIST, ZERO LEN ADDR
           7E  20 A8    7D  015C          MOVQ     EQUNAM(R8),-(SP)                 ;SET THE EQUIV NAME DESCR
           02 AE  02    B0  0160          MOVW     #LNM$_STRING,2(SP)               ;SET THE ITEM TYPE
              57  5E    D0  0164          MOVL     SP,R7                            ;GET THE ITEM LIST ADDRESS
                            0167
                            0167          $CRELNM_S  ATTR=NAME_ATTR(R8),-           ;CREATE THE REQUESTED NAME
                            0167                   TABNAM=TABNAM(R8),-              ;
                            0167                   LOGNAM=LOGNAM(R8),-              ;
                            0167                   ACMODE=ACMODE(R8),-              ;
                            0167                   ITMLST=(R7)                      ;
                            017C
              5E  10    C0  017C          ADDL     #4*4,SP                          ;POP THE ITEM LIST
                            017F
                            017F    ;
                            017F    ; Output informational message if appropriate.
                            017F    ;
        50   0631 8F    B1  017F          CMPW     #SS$_SUPERSEDE,R0                ;PREVIOUS ASSIGNMENT SUPERSEDED?
                  1C    12  0184          BNEQ     96$                              ;IF NEQ NO
        10 08 A8  00    E1  0186          BBC      #LOG_V_QUAL(R8),95$              ;BRANCH IF /NOLOG
              18 A8     9F  018B          PUSHAB   LOGNAM(R8)                       ;SET LOGICAL NAME ADDRESS
              51  01    D0  018E          MOVL     #1,R1                            ;SET FAO COUNT
    50   0003DDEB 8F    D0  0191          MOVL     #CLI$_SUPERSEDE,R0               ;SET STATUS
               FE65'    30  0198          BSBW     DCL$FORMMSG                      ;OUTPUT MESSAGE
                            019B
                            019B    95$:  STATUS   NORMAL                           ;RETURN SUCCESS
                        05  01A2    96$:  RSB                                       ;EXIT
```

```
                            01A3            .SBTTL  ASSIGN LOGICAL NAME TO EQUIVALENCE STRING
                            01A3       ;+
                            01A3       ; DCL$ASSIGN - ASSIGN LOGICAL NAME TO EQUIVALENCE STRING
                            01A3       ;
                            01A3       ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE ASSIGN
                            01A3       ; COMMAND.
                            01A3       ;
                            01A3       ; INPUTS:
                            01A3       ;
                            01A3       ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                            01A3       ;       R9 = ADDRESS OF SCRATCH STACK.
                            01A3       ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                            01A3       ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                            01A3       ;
                            01A3       ; OUTPUTS:
                            01A3       ;
                            01A3       ;       THE SPECIFIED LOGICAL NAME IS ASSIGNED TO THE SPECIFIED EQUIVALENCE
                            01A3       ;       STRING. IF A PREVIOUS LOGICAL ASSIGNMENT IS SUPERSEDED, THEN A
                            01A3       ;       MESSAGE TO THAT EFFECT IS WRITTEN TO THE OUTPUT STREAM.
                            01A3       ;-
                            01A3
                            01A3       DCL$ASSIGN::                                            ;ASSIGN LOGICAL NAME TO EQUIVALENCE
                            01A3       ;
                            01A3       ; Parse the common qualifiers and the logical name string.
                            01A3       ;
              0221    30    01A3            BSBW    COMMON_QUAL                     ;PROCESS COMMON QUALIFIERS
                79    7C    01A6            CLRQ    -(R9)                           ;ALLOCATE SPACE FOR EQUIV NAME
        51   18 A8    7D    01A8            MOVQ    LOGNAM(R8),R1                   ;GET EQUIV NAME DESCR
              FE51'   30    01AC            BSBW    DCL$COMPRESS                    ;COMPRESS THE STRING
        20 A8   51    7D    01AF            MOVQ    R1,EQUNAM(R8)                   ;SAVE EQUIV NAME
                            01B3
                            01B3       ;
                            01B3       ; Init the item list.  Insert the default translation attributes.
                            01B3       ;
      5E    F9FC CE   9E    01B3            MOVAB   -64*6*4-4(SP),SP                ;ALLOCATE ROOM FOR A 128 ITEM LIST
             57  5E   D0    01B8            MOVL    SP,R7                           ;SAVE THE ADDRESS OF THE LIST
                            01BB
    87   00030004 8F  D0    01BB            MOVL    #LNM$_ATTRIBUTES@16+4,(R7)+     ;SET THE ITEM TYPE
             87  68   DE    01C2            MOVAL   TRAN_ATTR(R8),(R7)+             ;SET THE DEFAULT ATTRIBUTES ADDR
                 87   D4    01C5            CLRL    (R7)+                           ;ZERO THE RETURN LENGTH ADDR
             56  01   D0    01C7            MOVL    #1,R6                           ;MARK DEFAULT ATTRIBUTES SET
                            01CA
                            01CA       ;
                            01CA       ; Loop getting equivalence strings and their attributes.
                            01CA       ; Build the item list.
                            01CA       ;
              0377    30    01CA       25$:  BSBW    GET_TRAN_ATTR                   ;CHECK FOR NEW TRAN ATTRIBUTES
             13  50   E9    01CD            BLBC    R0,30$                          ;BRANCH IF NO LOCAL QUALIFIER
    87   00030004 8F  D0    01D0            MOVL    #LNM$_ATTRIBUTES@16+4,(R7)+     ;SET THE ITEM TYPE
             79  53   D0    01D7            MOVL    R3,-(R9)                        ;SAVE THE ATTRIBUTES
             87  59   D0    01DA            MOVL    R9,(R7)+                        ;SET THE ATTRIBUTES ADDR
                 87   D4    01DD            CLRL    (R7)+                           ;ZERO THE RETURN LENGTH ADDR
                 56   D4    01DF            CLRL    R6                              ;MARK NEW ATTRIBUTES SET
                 12   11    01E1            BRB     40$                             ;PROCESS THE PARAMETER
             0F  56   E8    01E3       30$:  BLBS    R6,40$                          ;SKIP IF DEFAULTS IN EFFECT
    87   00030004 8F  D0    01E6            MOVL    #LNM$_ATTRIBUTES@16+4,(R7)+     ;SET THE ITEM TYPE
             87  68   DE    01ED            MOVAL   TRAN_ATTR(R8),(R7)+             ;SET THE DEFAULT ATTRIBUTES ADDR
```

```
          87  D4  01F0          CLRL    (R7)+                         ;ZERO THE RETURN LENGTH ADDR
      56  01  D0  01F2          MOVL    #1,R6                         ;MARK NEW ATTRIBUTES SET
                  01F5
 87   20  A8  7D  01F5  40$:    MOVQ    EQUNAM(R8),(R7)+              ;SAVE THE EQUIV NAME DESCR
 FA  A7   02  B0  01F9          MOVW    #LNM$_STRING,-6(R7)           ;SET THE ITEM TYPE
          87  D4  01FD          CLRL    (R7)+                         ;ZERO THE RETURN LENGTH ADDR
          0C  C3  01FF          SUBL3   #PTR_K_LENGTH,-               ;GET ADDRESS OF TOKEN DESCRIPTOR
 50   BA  AA      0201                  WRK_L_RSLNXT(R10),R0          ;
 02   06  A0  91  0204          CMPB    PTR_B_PARMCNT(R0),#2          ;HAVE WE FOUND THE LOGICAL NAME?
          09  13  0208          BEQL    43$                           ;YES, THEN TERMINATE ITEM LIST
       FDF3'  30  020A          BSBW    DCL$COMPRESS                  ;COMPRESS THE STRING
   20 A8  51  7D  020D          MOVQ    R1,EQUNAM(R8)                 ;STORE THE LATEST STRING DESCR
          B7  11  0211          BRB     25$                           ;CHECK FOR NEW ATTRIBUTES
                  0213
          87  D4  0213  43$:    CLRL    (R7)+                         ;TERMINATE THE LIST
                  0215
                  0215  ;
                  0215  ; Process logical name string.
                  0215  ;
       FDE8'  30  0215          BSBW    DCL$COMPSTRING                ;REMOVE QUOTATION MARKS
 FF A241  3A  91  0218          CMPB    #^A/:/,-1(R2)[R1]             ;LOGICAL NAME END WITH COLON?
          02  12  021D          BNEQ    45$                           ;IF NEQ NO
          51  D7  021F          DECL    R1                            ;REDUCE LENGTH OF LOGICAL NAME
   18 A8  51  7D  0221  45$:    MOVQ    R1,LOGNAM(R8)                 ;SAVE LOGICAL NAME DESCRIPTOR
        031C  30  0225          BSBW    GET_TRAN_ATTR                 ;CHECK FOR NEW TRAN ATTRIBUTES
       03 50  E9  0228          BLBC    R0,47$                        ;BRANCH IF NO LOCAL QUALIFIER
      68  53  D0  022B          MOVL    R3,TRAN_ATTR(R8)              ;SAVE THE ATTRIBUTES
        007C  31  022E  47$:    BRW     COMMON_CRELNM                 ;CREATE THE LOGICAL NAME
```

```
                         0231          .SBTTL  DEFINE LOGICAL NAME EQUIVALENCE
                         0231      ;+
                         0231      ; DCL$DEFINE - DEFINE LOGICAL NAME EQUIVALENCE
                         0231      ;
                         0231      ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE DEFINE
                         0231      ; COMMAND.
                         0231      ;
                         0231      ; INPUTS:
                         0231      ;
                         0231      ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                         0231      ;       R9 = ADDRESS OF SCRATCH STACK.
                         0231      ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                         0231      ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                         0231      ;
                         0231      ; OUTPUTS:
                         0231      ;
                         0231      ;       THE SPECIFIED LOGICAL NAME IS ASSIGNED TO THE SPECIFIED EQUIVALENCE
                         0231      ;       STRING. IF A PREVIOUS LOGICAL ASSIGNMENT IS SUPERSEDED, THEN A
                         0231      ;       MESSAGE TO THAT EFFECT IS WRITTEN TO THE OUTPUT STREAM.
                         0231      ;-
                         0231
                         0231      DCL$DEFINE::                                         ;DEFINE LOGICAL NAME EQUIVALENCE
                         0231      ;
                         0231      ; Parse the common qualifiers and the logical name string.
                         0231      ;
            0193   30    0231             BSBW    COMMON_QUAL                          ;PROCESS COMMON QUALIFIERS
              79   7C    0234             CLRQ    -(R9)                                ;ALLOCATE SPACE FOR EQUIVALENCE NAME
     51    18 A8   7D    0236             MOVQ    LOGNAM(R8),R1                        ;GET LOGICAL NAME DESCR
           FDC3'   30    023A             BSBW    DCL$COMPSTRING                       ;REMOVE QUOTES FROM LOGICAL NAME
     18 A8    51   7D    023D             MOVQ    R1,LOGNAM(R8)                        ;SAVE LOGICAL NAME
                         0241
                         0241      ;
                         0241      ; Init the item list.  Insert the default translation attributes.
                         0241      ;
     5E    F9FC CE 9E    0241             MOVAB   -64*6*4-4(SP),SP                     ;ALLOCATE ROOM FOR A 128 ITEM LIST
         57     5E D0    0246             MOVL    SP,R7                                ;SAVE THE ADDRESS OF THE LIST
                         0249
  87    00030004 8F D0   0249             MOVL    #LNM$_ATTRIBUTES@16+4,(R7)+          ;SET THE ITEM TYPE
         87     68 DE    0250             MOVAL   TRAN_ATTR(R8),(R7)+                  ;SET THE DEFAULT ATTRIBUTES ADDR
                87 D4    0253             CLRL    (R7)+                                ;ZERO THE RETURN LENGTH ADDR
                         0255
            02EC   30    0255             BSBW    GET_TRAN_ATTR                        ;CHECK FOR NEW TRAN ATTRIBUTES
           03 50   E9    0258             BLBC    R0,23$                               ;BRANCH IF NO LOCAL QUALIFIER
         68   53   D0    025B             MOVL    R3,TRAN_ATTR(R8)                     ;SAVE THE ATTRIBUTES
           FD9F'   30    025E      23$:   BSBW    DCL$COMPRESS                         ;COMPRESS THE STRING
     20 A8    51   7D    0261             MOVQ    R1,EQUNAM(R8)                        ;LOAD THE PIPELINE
         56   01   D0    0265             MOVL    #1,R6                                ;MARK DEFAULT ATTRIBUTES SET
                         0268
                         0268      ;
                         0268      ; Loop getting equivalence strings and their attributes.
                         0268      ; Build the item list.
                         0268      ;
            02D9   30    0268      25$:   BSBW    GET_TRAN_ATTR                        ;CHECK FOR NEW TRAN ATTRIBUTES
           13 50   E9    026B             BLBC    R0,30$                               ;BRANCH IF NO LOCAL QUALIFIER
  87    00030004 8F D0   026E             MOVL    #LNM$_ATTRIBUTES@16+4,(R7)+          ;SET THE ITEM TYPE
         79   53   D0    0275             MOVL    R3,-(R9)                             ;SAVE THE ATTRIBUTES
         87   59   D0    0278             MOVL    R9,(R7)+                             ;SET THE ATTRIBUTES ADDR
```

```
                 87   D4  027B           CLRL    (R7)+                           ;ZERO THE RETURN LENGTH ADDR
                 56   D4  027D           CLRL    R6                              ;MARK NEW ATTRIBUTES SET
                 12   11  027F           BRB     40$                             ;PROCESS THE PARAMETER
              0F 56   E8  0281     30$:  BLBS    R6,40$                          ;SKIP IF DEFAULTS IN EFFECT
   87   00030004 8F   D0  0284           MOVL    #LNM$_ATTRIBUTES@16+4,(R7)+     ;SET THE ITEM TYPE
            87   68   DE  028B           MOVAL   TRAN_ATTR(R8),(R7)+             ;SET THE DEFAULT ATTRIBUTES ADDR
                 87   D4  028E           CLRL    (R7)+                           ;ZERO THE RETURN LENGTH ADDR
            56   01   D0  0290           MOVL    #1,R6                           ;MARK NEW ATTRIBUTES SET
                          0293
        87    20 A8   7D  0293     40$:  MOVQ    EQUNAM(R8),(R7)+                ;SAVE THE EQUIV NAME DESCR
        FA A7    02   B0  0297           MOVW    #LNM$_STRING,-6(R7)             ;SET THE ITEM TYPE
                 87   D4  029B           CLRL    (R7)+                           ;ZERO THE RETURN LENGTH ADDR
            55   04   91  029D           CMPB    #PTR_K_ENDLINE,R5               ;EOL?
                 09   13  02A0           BEQL    43$                             ;YES, THEN TERMINATE ITEM LIST
              FD5B'   30  02A2           BSBW    DCL$COMPRESS                    ;COMPRESS THE STRING
        20 A8    51   7D  02A5           MOVQ    R1,EQUNAM(R8)                   ;STORE THE LATEST STRING DESCR
                 BD   11  02A9           BRB     25$                             ;CHECK FOR NEW ATTRIBUTES
                          02AB
                 87   D4  02AB     43$:  CLRL    (R7)+                           ;TERMINATE THE LIST
                          02AD
```

```
                             02AD      ;
                             02AD      ; Check for SYS$OUTPUT.  Do special processing if appropriate.
                             02AD      ;
                             02AD      COMMON_CRELNM:
           57    1C     C2   02AD              SUBL    #6*4+4,R7                        ;WAS MORE THAN ONE VALUE SUPPLIED
           5E    57     D1   02B0              CMPL    R7,SP                            ;
                 05     12   02B3              BNEQ    45$                              ;YES, THEN BRANCH
               00D9     30   02B5              BSBW    TESTOUT                          ;IS LOGICAL NAME SYS$OUTPUT?
                 46     13   02B8              BEQL    80$                              ;YES, THEN BRANCH
                             02BA
                             02BA      ;
                             02BA      ; Create the requested logial names.
                             02BA      ;
    00 08 A8     01     E1   02BA      45$:    BBC     #ATTR_V,QUAL(R8),47$             ;BRANCH IF QUALIFIER NOT SEEN
                             02BF      ;       BICL    #LNM$M_CRELOG,NAME_ATTR(R8)      ;DISABLE CRELOG ATTRIBUTE
           57    5E     D0   02BF      47$:    MOVL    SP,R7                            ;GET THE ITEM LIST ADDRESS
                             02C2              $CRELNM_S  ATTR=NAME_ATTR(R8),-          ;CREATE THE REQUESTED NAME
                             02C2                      TABNAM=TABNAM(R8),-              ;
                             02C2                      LOGNAM=LOGNAM(R8),-              ;
                             02C2                      ACMODE=ACMODE(R8),-              ;
                             02C2                      ITMLST=(R7)                      ;
                             02D7
                             02D7      ;
                             02D7      ; Output informational message if appropriate.
                             02D7      ;
    50     0631 8F      B1   02D7              CMPW    #SS$_SUPERSEDE,R0                ;PREVIOUS ASSIGNMENT SUPERSEDED?
                 1C     12   02DC              BNEQ    60$                              ;IF NEQ NO
    10 08 A8     00     E1   02DE              BBC     #LOG_V,QUAL(R8),50$              ;BRANCH IF /NOLOG
              18 A8     9F   02E3              PUSHAB  LOGNAM(R8)                       ;SET LOGICAL NAME ADDRESS
           51    01     D0   02E6              MOVL    #1,R1                            ;SET FAO COUNT
50    0003DDEB 8F       D0   02E9              MOVL    #CLI$_SUPERSEDE,R0               ;SET STATUS
              FD0D'     30   02F0              BSBW    DCL$FORMMSG                      ;OUTPUT MESSAGE
                             02F3
                             02F3      50$:    STATUS  NORMAL                           ;SET NORMAL COMPLETION
    5E     0604 CE      9E   02FA      60$:    MOVAB   64*6*4+4(SP),SP                  ;RESTORE THE STACK
                        05   02FF              RSB                                      ;
                             0300
                             0300      ;
                             0300      ; Update SYS$OUTPUT.
                             0300      ;
      51     20 A8      7D   0300      80$:    MOVQ    EQUNAM(R8),R1                    ;GET DESCRIPTOR OF EQUIVALENCE NAME
              FCF9'     30   0304              BSBW    DCL$OPEN_OUTPUT                  ;OPEN SPECIFIED OUTPUT FILE
              F0 50     E9   0307              BLBC    R0,60$                           ;LEAVE EVERYTHING ALONE IF ERROR
      58   00BC CB      D0   030A              MOVL    PRC_L_IDFLNK(R11),R8             ;POINT TO THE SYS$OUTPUT INFORMATION
              FCEE'     30   030F              BSBW    DCL$CREATE_OUTPUT                ;CREATE THE SYS$OUTPUT LOGICAL NAME
                 DF     11   0312              BRB     50$                              ;
```

```
                  0314          .SBTTL  DEALLOCATE DEVICE
                  0314     ;+
                  0314     ; DCL$DEALLOCAT - DEALLOCATE DEVICE
                  0314     ;
                  0314     ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE DEALLOCATE
                  0314     ; DCL$ COMMAND.
                  0314     ;
                  0314     ; INPUTS:
                  0314     ;
                  0314     ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                  0314     ;       R9 = ADDRESS OF SCRATCH STACK.
                  0314     ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                  0314     ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                  0314     ;
                  0314     ; OUTPUTS:
                  0314     ;
                  0314     ;       THE SPECIFIED DEVICE IS DEALLOCATED OR ALL DEVICES ARE DEALLOCATED.
                  0314     ;-
                  0314
                  0314     DCL$DEALLOCAT::                                     ;DEALLOCATE DEVICE
       FCE9'  30  0314             BSBW    DCL$GETDVAL                 ;GET TOKEN DESCRIPTOR
    55   03   91  0317             CMPB    #PTR_K_PARAMETR,R5          ;ITEM TYPE PARAMETER?
         04   13  031A             BEQL    10$                         ;YES, PROCESS IT
         59   D4  031C             CLRL    R9                          ;NO, ASSUME /ALL
         0F   11  031E             BRB     90$                         ;DEALLOCATE THEM ALL
       FCDD'  30  0320     10$:    BSBW    DCL$COMPSTRING              ;REMOVE EXTERNAL QUOTATION MARKS
    79   51   7D  0323             MOVQ    R1,-(R9)                    ;SAVE LOGICAL NAME
FF A241   3A   91  0326             CMPB    #^A/:/,-1(R2)[R1]           ;STRING END WITH A COLON
         02   12  032B             BNEQ    90$                         ;BR IF NO
         69   D7  032D             DECL    (R9)                        ;REMOVE COLON FROM STRING
              032F     90$:    $DALLOC_S (R9)              ;DEALLOCATE DEVICE
          05  033A             RSB                                 ;
```

```
                         033B          .SBTTL  DEASSIGN LOGICAL NAME EQUIVALENCE
                         033B  ;+
                         033B  ; DCL$DEASSIGN - DEASSIGN LOGICAL NAME EQUIVALENCE
                         033B  ;
                         033B  ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE DEASSIGN DCLS
                         033B  ; COMMAND.
                         033B  ;
                         033B  ; INPUTS:
                         033B  ;
                         033B  ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                         033B  ;       R9 = ADDRESS OF SCRATCH STACK.
                         033B  ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                         033B  ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                         033B  ;
                         033B  ; OUTPUTS:
                         033B  ;
                         033B  ;       THE SPECIFIED LOGICAL NAME EQUIVALENCE OR ALL LOGICAL NAME EQUIVALENCES
                         033B  ;       ARE DEASSIGNED.
                         033B  ;-
                         033B
                         033B  DCL$DEASSIGN::                                   ;DEASSIGN LOGICAL NAME EQUIVALENCE
             0089   30   033B         BSBW    COMMON_QUAL                      ;PROCESS COMMON QUALIFIERS
               57   D4   033E         CLRL    R7                               ;ASSUME DOING /ALL
    05 08 A8   02   E1   0340         BBC     #DEF_V,QUAL(R8),3$               ;SKIP IF DEFAULTED
    19 08 A8   00   E0   0345         BBS     #LOG_V,QUAL(R8),5$               ;BR IF DOING /ALL
      51  18 A8   7D   034A  3$:     MOVQ    LOGNAM(R8),R1                    ;GET LOGICAL NAME
           FCAF'   30   034E         BSBW    DCL$COMPSTRING                   ;REMOVE EXTERNAL QUOTATION MARKS
       18 A8   51   7D   0351         MOVQ    R1,LOGNAM(R8)                    ;SAVE LOGICAL NAME
      57  18 A8   7E   0355         MOVAQ   LOGNAM(R8),R7                    ;COPY THE DESCRIPTOR ADDRESS
    FF A241   3A   91   0359         CMPB    #^A/:/,-1(R2)[R1]                ;STRING END WITH A COLON
               03   12   035E         BNEQ    5$                               ;BR IF NO
            18 A8   D7   0360         DECL    LOGNAM(R8)                       ;REMOVE COLON FROM STRING
               2C   10   0363  5$:     BSBB    TESTOUT                          ;IS LOGICAL NAME SYS$OUTPUT?
               14   13   0365         BEQL    10$                              ;YES, THEN SKIP
                         0367         $DELLNM_S  TABNAM=TABNAM(R8),-           ;DEASSIGN LOGICAL NAME EQUIVALENCE
                         0367                 LOGNAM=(R7),-                    ;
                         0367                 ACMODE=ACMODE(R8)                ;
               57   D5   0376         TSTL    R7                               ;DEASSIGN/ALL?
               0E   13   0378         BEQL    20$                              ;YES, THEN RECREATE SYS$OUTPUT
                    05   037A         RSB                                      ;
                         037B
    52   0114 CB   9E   037B  10$:    MOVAB   PRC_W_OUTIFI(R11),R2             ;GET ADDRESS OF SYS$OUTPUT INFORMATION
    58   00BC CB   D0   0380         MOVL    PRC_L_IDFLNK(R11),R8             ;GET ADDRESS OF CURRENT IDF BLOCK
           FC78'   30   0385         BSBW    DCL$RESTORE_OUTPUT               ;RESTORE PROCESS PERMANENT SYS$OUTPUT
    58   00BC CB   D0   0388  20$:    MOVL    PRC_L_IDFLNK(R11),R8             ;GET ADDRESS OF CURRENT IDF BLOCK
           FC70'   30   038D         BSBW    DCL$CREATE_OUTPUT                ;CREATE SYS$OUTPUT LOGICAL NAME
                    05   0390         RSB
```

```
                         0391          .SBTTL  TEST IF LOGICAL NAME IS SYS$OUTPUT
                         0391     ;
                         0391     ; SUBROUTINE TO TEST IF LOGICAL NAME IS SYS$OUTPUT
                         0391     ;
                         0391     ; ON OUTPUT, 'Z'= 1 IF SYS$OUTPUT IS SPECIFIED
                         0391     ;
                         0391     ; CLOBBERS R0-R3
                         0391     ;
                         0391     TESTOUT:
          18 A8     D5   0391          TSTL    LOGNAM(R8)                 ;WAS A LOGICAL NAME SPECIFIED?
             2E     13   0394          BEQL    20$                        ;RETURN IF NOT
    29 08 A8        01   E0   0396     BBS     #ATTR_V_QUAL(R8),20$       ;BRANCH IF ATTRIBUTES SPECIFIED
       0C A8        03   D1   039B     CMPL    #PSL$C_USER,ACMODE(R8)     ;IS LOGICAL NAME USER MODE?
             23     13   039F          BEQL    20$                        ;YES, THEN RETURN
  10 A8    FC66 CF  91   03A1          CMPB    LNM$PROCESS,TABNAM(R8)     ;COMPARE LENGTH OF TABLE NAME
             1B     12   03A7          BNEQ    20$                        ;RETURN IF NOT EQUAL
    14 B8    10 A8  29   03A9          CMPC    TABNAM(R8),@TABNAM+4(R8),-;COMPARE ACTUAL STRING
         FC5B CF         03AE                  LNM$PROCESS+1              ;
             11     12   03B1          BNEQ    20$                        ;RETURN IF NOT EQUAL
  18 A8    FC49 CF  91   03B3          CMPB    OUTPUTNAM,LOGNAM(R8)       ;COMPARE LENGTH OF OUTPUT
             09     12   03B9          BNEQ    20$                        ;RETURN IF NOT EQUAL
    1C B8    18 A8  29   03BB          CMPC    LOGNAM(R8),@LOGNAM+4(R8),-;COMPARE ACTUAL STRING
         FC3E CF         03C0                  OUTPUTNAM+1                ;
                    05   03C3     10$:  RSB                               ;
                         03C4
             01     D5   03C4     20$:  TSTL    #1                        ;SET FAILURE STATUS
                    05   03C6          RSB                                ;
```

```
                          03C7   591          .SBTTL  PROCESS COMMON COMMAND QUALIFIERS
                          03C7   592 ;
                          03C7   593 ; SUBROUTINE TO PROCESS COMMON COMMAND QUALIFIERS
                          03C7   594 ;
                          03C7   595 ; ON INPUT,  R9 = ADDRESS OF SCRATCH STACK
                          03C7   596 ;
                          03C7   597 ; ON OUTPUT, SCRATCH STACK LOOKS LIKE
                          03C7   598 ;
                          03C7   599 ;                             <-- R9 initially
                          03C7   600 ;             ;---------------;
                          03C7   601 ;             ; Logical name  ; LOGNAM(R8)
                          03C7   602 ;             ;  descriptor   ;
                          03C7   603 ;             ;---------------;
                          03C7   604 ;             ;  Table name   ; TABNAM(R8)
                          03C7   605 ;             ;  descriptor   ;
                          03C7   606 ;             ;---------------;
                          03C7   607 ;             ;  Access mode  ; ACMODE(R8)
                          03C7   608 ;             ;---------------;
                          03C7   609 ;             ; Qual flags    ; QUAL(R8)
                          03C7   610 ;             ;---------------;
                          03C7   611 ;             ; Def name attr ; NAME_ATTR(R8)
                          03C7   612 ;             ;---------------;
                          03C7   613 ;             ; Def tran attr ; TRAN_ATTR(R8) <-- R8,R9 finally
                          03C7   614 ;             ;---------------;
                          03C7   615 ;
                          03C7   616 ;
                          03C7   617 COMMON_QUAL:
             79     7C    03C7   618          CLRQ    -(R9)                           ;ALLOCATE SPACE FOR LOG NAME DESCR
    55  FC3E CF     9E    03C9   619          MOVAB   LNM$PROCESS,R5                  ;ASSUME PROCESS LOGICAL NAME TABLE
        54   85     9A    03CE   620          MOVZBL  (R5)+,R4                        ;
        79   54     7D    03D1   621          MOVQ    R4,-(R9)                        ;SAVE THE DESCRIPTOR
        79   02     D0    03D4   622          MOVL    #PSL$C_SUPER,-(R9)              ;AND SUPERVISOR MODE
        79   01     D0    03D7   623          MOVL    #LOG_M,-(R9)                    ;ASSUME /LOG DEFAULTED
                          03DA   624 ;        MOVL    #LNM$M_CRELOG,-(R9)             ;SET DEFAULT NAME ATTRIBUTES
             79     D4    03DA   625          CLRL    -(R9)                           ;SET DEFAULT NAME ATTRIBUTES
             79     D4    03DC   626          CLRL    -(R9)                           ;SET DEFAULT TRAN ATTRIBUTES
        58   59     D0    03DE   627          MOVL    R9,R8                           ;COPY THE BASE OF THE DATA STRUCTURE
                          03E1   628
          FC1C'     30    03E1   629 10$:     BSBW    DCL$GETDVAL                     ;GET NEXT DESCRIPTOR VALUE
        55   03     91    03E4   630 20$:     CMPB    #PTR_K_PARAMETR,R5              ;ITEM TYPE PARAMETER OR EOL?
             08     1A    03E7   631          BGTRU   30$                             ;NO, THEN PROCESS QUALIFIER
             03     13    03E9   632          BEQL    25$                             ;PROCESS PARAMETER
           0155     31    03EB   633          BRW     210$                            ;RETURN WITH NULL PARAMETER
           014E     31    03EE   634 25$:     BRW     200$                            ;PROCESS PARAMETER
          FC0C'     30    03F1   635 30$:     BSBW    DCL$GETNVAL                     ;GET QUALIFIER NUMBER
00000000'8F  51     D1    03F4   636          CMPL    R1,#CLI$K_DEFI_USER             ;QUALIFIER MATCH?
             66     13    03FB   637          BEQL    100$                            ;YES, THEN BRANCH
00000000'8F  51     D1    03FD   638          CMPL    R1,#CLI$K_DEFI_SUPE             ;QUALIFIER MATCH?
             64     13    0404   639          BEQL    110$                            ;YES, THEN BRANCH
00000000'8F  51     D1    0406   640          CMPL    R1,#CLI$K_DEFI_EXEC             ;QUALIFIER MATCH?
             61     13    040D   641          BEQL    120$                            ;YES, THEN BRANCH
00000000'8F  51     D1    040F   642          CMPL    R1,#CLI$K_DEFI_PROC             ;QUALIFIER MATCH?
             5E     13    0416   643          BEQL    130$                            ;YES, THEN BRANCH
00000000'8F  51     D1    0418   644          CMPL    r1,#CLI$K_DEFI_JOB              ;QUALIFIER MATCH?
             63     13    041F   645          BEQL    135$                            ;YES, THEN BRANCH
00000000'8F  51     D1    0421   646          CMPL    R1,#CLI$K_DEFI_GROU             ;QUALIFIER MATCH?
             71     13    0428   647          BEQL    140$                            ;YES, THEN BRANCH
```

```
00000000'8F   51  D1 042A          CMPL    R1,#CLI$K_DEFI_SYST            ;QUALIFIER MATCH?
              7F  13 0431          BEQL    150$                           ;YES, THEN BRANCH
00000000'8F   51  D1 0433          CMPL    R1,#CLI$K_DEFI_TABL            ;QUALIFIER MATCH?
              03  12 043A          BNEQ    60$                            ;NO, CHECK NEXT
            008A  31 043C          BRW     160$                           ;YES, THEN BRANCH
00000000'8F   51  D1 043F  60$:    CMPL    R1,#CLI$K_DEFI_LOG             ;QUALIFIER MATCH? (ALSO DEASSIGN/ALL
              03  12 0446          BNEQ    70$                            ;NO, CHECK NEXT
            0096  31 0448          BRW     170$                           ;YES, THEN BRANCH
00000000'8F   51  D1 044B  70$:    CMPL    R1,#CLI$K_DEFI_NAME            ;QUALIFIER MATCH?
              03  12 0452          BNEQ    80$                            ;NO, CHECK NEXT
            0C9B  31 0454          BRW     180$                           ;YES, THEN BRANCH
00000000'8F   51  D1 0457  80$:    CMPL    R1,#CLI$K_DEFI_TRAN            ;QUALIFIER MATCH?
              81  12 045E          BNEQ    10$                            ;NO, IGNORE IT
            00B3  31 0460          BRW     190$                           ;YES, THEN BRANCH
                     0463
      0C A8   03  D0 0463  100$:   MOVL    #PSL$C_USER,ACMODE(R8)         ;SET USER MODE
            0082  31 0467          BRW     171$                           ;GET NEXT TOKEN
      0C A8   02  D0 046A  110$:   MOVL    #PSL$C_SUPER,ACMODE(R8)        ;SET SUPER MODE
              7C  11 046E          BRB     171$                           ;GET NEXT TOKEN
      0C A8   01  D0 0470  120$:   MOVL    #PSL$C_EXEC,ACMODE(R8)         ;SET EXEC MODE
              76  11 0474          BRB     171$                           ;GET NEXT TOKEN
                     0476
   55   FB91 CF   9E 0476  130$:   MOVAB   LNM$PROCESS,R5                 ;USE PROCESS LOGICAL NAME TABLE
          54  85  9A 047B          MOVZBL  (R5)+,R4                       ;
     10 A8    54  7D 047E          MOVQ    R4,TABNAM(R8)                  ;SAVE THE DESCRIPTOR
              68  11 0482          BRB     171$                           ;GET NEXT TOKEN
                     0484
   55   FB83 CF   9E 0484  135$:   MOVAB   LNM$PROCESS,R5                 ;ASSUME /NOJOB
    05 53     00  E0 0489          BBS     #PTR_V_NEGATE-PTR_V_FLAGS,R3,137$  ;BR IF /NOJOB
   55   FB86 CF   9E 048D          MOVAB   LNM$JOB,R5                     ;USE JOB LOGICAL NAME TABLE
          54  85  9A 0492  137$:   MOVZBL  (R5)+,R4                       ;
     10 A8    54  7D 0495          MOVQ    R4,TABNAM(R8)                  ;SAVE THE DESCRIPTOR
              51  11 0499          BRB     171$                           ;GET NEXT TOKEN
                     049B
   55   FB6C CF   9E 049B  140$:   MOVAB   LNM$PROCESS,R5                 ;ASSUME /NOGROUP
    05 53     00  E0 04A0          BBS     #PTR_V_NEGATE-PTR_V_FLAGS,R3,147$  ;BR IF /NOGROUP
   55   FB77 CF   9E 04A4          MOVAB   LNM$GROUP,R5                   ;USE GROUP LOGICAL NAME TABLE
          54  85  9A 04A9  147$:   MOVZBL  (R5)+,R4                       ;
     10 A8    54  7D 04AC          MOVQ    R4,TABNAM(R8)                  ;SAVE THE DESCRIPTOR
              3A  11 04B0          BRB     171$                           ;GET NEXT TOKEN
                     04B2
   55   FB55 CF   9E 04B2  150$:   MOVAB   LNM$PROCESS,R5                 ;ASSUME /NOSYSTEM
    05 53     00  E0 04B7          BBS     #PTR_V_NEGATE-PTR_V_FLAGS,R3,157$  ;BR IF /NOSYSTEM
   55   FB6A CF   9E 04BB          MOVAB   LNM$SYSTEM,R5                  ;USE SYSTEM LOGICAL NAME TABLE
          54  85  9A 04C0  157$:   MOVZBL  (R5)+,R4                       ;
     10 A8    54  7D 04C3          MOVQ    R4,TABNAM(R8)                  ;SAVE THE DESCRIPTOR
              23  11 04C7          BRB     171$                           ;GET NEXT TOKEN
                     04C9
   55   FB3E CF   9E 04C9  160$:   MOVAB   LNM$PROCESS,R5                 ;ASSUME PROCESS LOGICAL NAME TABLE
          54  85  9A 04CE          MOVZBL  (R5)+,R4                       ;
     10 A8    54  7D 04D1          MOVQ    R4,TABNAM(R8)                  ;SAVE THE DESCRIPTOR
           14 53  E8 04D5          BLBS    R3,171$                        ;BRANCH IF NEGATED
           FB25'  30 04D8          BSBW    DCL$GETDVAL                    ;GET THE TABLE NAME
     10 A8    51  7D 04DB          MOVQ    R1,TABNAM(R8)                  ;SAVE IT AWAY
              0B  11 04DF          BRB     171$                           ;GET NEXT TOKEN
                     04E1
     08 A8    05  C8 04E1  170$:   BISL    #LOG_M!DEF_M,QUAL(R8)          ;ASSUME /LOG OR /ALL
```

```
             04 53   E9  04E5    705         BLBC    R3,171$                           ;BRANCH IF SO
       08 A8    01   CA  04E8    706         BICL    #LOG_M,QUAL(R8)                   ;SET /NOLOG OR /NOALL
              FEF2   31  04EC    707 171$:   BRW     10$                               ;GET NEXT TOKEN
              FEF2   31  04EF    708 172$:   BRW     20$                               ;PROCESS THE TOKEN
                         04F2    709
       08 A8    02   C8  04F2    710 180$:   BISL    #ATTR_M,QUAL(R8)                  ;MARK /NAME_ATTRIBUTES SEEN
             04 A8   D4  04F6    711         CLRL    NAME_ATTR(R8)                     ;ZERO INITIAL ATTRIBUTES
             F0 53   E8  04F9    712         BLBS    R3,171$                           ;BRANCH IF NEGATED
             FB01'   30  04FC    713 182$:   BSBW    DCL$GETDVAL                       ;GET ITS VALUE
          02    55   91  04FF    714         CMPB    R5,#PTR_K_QUALVALU                ;SKIP IF NOT A QUALIFIER VALUE
                EB   12  0502    715         BNEQ    172$                              ;
       43 8F    62   91  0504    716         CMPB    (R2),#^A/C/                       ;CONFINE KEYWORD?
                06   12  0508    717         BNEQ    184$                              ;NO, THEN BRANCH
       04 A8    02   C8  050A    718         BISL    #LNM$M_CONFINE,NAME_ATTR(R8)      ;SET THE ATTRIBUTE
                EC   11  050E    719         BRB     182$                              ;GET NEXT VALUE
       04 A8    01   C8  0510    720 184$:   BISL    #LNM$M_NO_ALIAS,NAME_ATTR(R8)     ;SET THE ATTRIBUTE
                E6   11  0514    721         BRB     182$                              ;GET NEXT VALUE
                         0516    722
       08 A8    02   C8  0516    723 190$:   BISL    #ATTR_M,QUAL(R8)                  ;MARK /TRANSLATION_ATTRIBUTES SEEN
                68   D4  051A    724         CLRL    TRAN_ATTR(R8)                     ;ZERO INITIAL ATTRIBUTES
             CD 53   E8  051C    725         BLBS    R3,171$                           ;BRANCH IF NEGATED
             FADE'   30  051F    726 192$:   BSBW    DCL$GETDVAL                       ;GET ITS VALUE
          02    55   91  0522    727         CMPB    R5,#PTR_K_QUALVALU                ;SKIP IF NOT A QUALIFIER VALUE
                C8   12  0525    728         BNEQ    172$                              ;
       43 8F    62   91  0527    729         CMPB    (R2),#^A/C/                       ;CONCEALED KEYWORD?
                09   12  052B    730         BNEQ    194$                              ;NO, THEN BRANCH
 68    00000100 8F   C8  052D    731         BISL    #LNM$M_CONCEALED,TRAN_ATTR(R8)    ;SET THE ATTRIBUTE
                E9   11  0534    732         BRB     192$                              ;GET NEXT VALUE
 68    00000200 8F   C8  0536    733 194$:   BISL    #LNM$M_TERMINAL,TRAN_ATTR(R8)     ;SET THE ATTRIBUTE
                E0   11  053D    734         BRB     192$                              ;GET NEXT VALUE
                         053F    735
       18 A8    51   7D  053F    736 200$:   MOVQ    R1,LOGNAM(R8)                     ;GET FIRST PARAMETER DESCRIPTOR
                     05  0543    737 210$:   RSB                                       ;
                         0544    738
                         0544   740             .SBTTL  GET TRANSLATION ATTRIBUTES
                         0544   741     ;
                         0544   742     ; SUBROUTINE TO PROCESS TRANSLATION ATTRIBUTES
                         0544   743     ;
                         0544   744     ;       R0 = QUALIFIER SEEN FLAG
                         0544   745     ;       R1/R2 = DESCR OF NEXT PARAMETER
                         0544   746     ;       R3 = TRANSLATION ATTRIBUTES
                         0544   747     ;       R5 = TYPE OF LAST TOKEN SEEN
                         0544   748     ;
                         0544   749 GET_TRAN_ATTR:                                      ;GET TRAN ATTRIBUTES
               7E   7C   0544   750             CLRQ    -(SP)                           ;ASSUME QUALIFIER NOT SEEN
            FAB7'   30   0546   751 10$:        BSBW    DCL$GETDVAL                     ;GET NEXT DESCRIPTOR VALUE
         55    03   91   0549   752 20$:        CMPB    #PTR_K_PARAMETR,R5              ;ITEM TYPE PARAMETER?
               2D   15   054C   753             BLEQ    90$                             ;IF LEQ END OF LINE OR PARAMETER
      04 AE    01   D0   054E   754             MOVL    #1,4(SP)                        ;MARK QUALIFIER SEEN
               6E   D4   0552   755             CLRL    (SP)                            ;RESET ATTRIBUTES
      08 A8    02   C8   0554   756             BISL    #ATTR_M,QUAL(R8)                ;MARK /TRANSLATION_ATTRIBUTES SEEN
            EB 53   E8   0558   757             BLBS    R3,10$                          ;BRANCH IF NEGATED
            FAA2'   30   055B   758 30$:        BSBW    DCL$GETDVAL                     ;GET ITS VALUE
         02    55   91   055E   759             CMPB    R5,#PTR_K_QUALVALU              ;SKIP IF NOT A QUALIFIER VALUE
               E6   12   0561   760             BNEQ    20$                             ;
      43 8F    62   91   0563   761             CMPB    (R2),#^A/C/                     ;CONCEALED KEYWORD?
               09   12   0567   762             BNEQ    40$                             ;NO, THEN BRANCH
6E   00000100 8F   C8   0569   763             BISL    #LNM$M_CONCEALED,(SP)           ;SET THE ATTRIBUTE
               E9   11   0570   764             BRB     30$                             ;GET NEXT VALUE
6E   00000200 8F   C8   0572   765 40$:        BISL    #LNM$M_TERMINAL,(SP)            ;SET THE ATTRIBUTE
               E0   11   0579   766             BRB     30$                             ;GET NEXT VALUE
                         057B   767
             53 8ED0     057B   768 90$:        POPL    R3                              ;RETURN ATTRIBUTES AND STATUS
             50 8ED0     057E   769             POPL    R0                              ;
                  05     0581   770             RSB                                     ;
```

```
                              0582          .SBTTL  CREATE LOGICAL NAME TABLE
                              0582      ;+
                              0582      ; DCL$CRETABLE - CREATE LOGICAL NAME TABLE
                              0582      ;
                              0582      ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE
                              0582      ; CREATE/NAME_TABLE COMMAND.
                              0582      ;
                              0582      ; INPUTS:
                              0582      ;
                              0582      ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                              0582      ;       R9 = ADDRESS OF SCRATCH STACK.
                              0582      ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                              0582      ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                              0582      ;
                              0582      ; OUTPUTS:
                              0582      ;
                              0582      ;       THE SPECIFIED LOGICAL NAME TABLE IS CREATED.
                              0582      ;-
                              0582
                              0582      DCL$CRETABLE::                                          ;CREATE A LOGICAL NAME TABLE
                79      D4    0582              CLRL    -(R9)                                   ;ASSUME /LOG
       79    FF00 8F    3C    0584              MOVZWL  #DEF_PROT,-(R9)                         ;SET DEFAULT TABLE PROTECTION
                              0589                                                              ;(SY:RWED,OW:RWED,GR,WO)
                              0589
            79   02     D0    0589              MOVL    #PSL$C_SUPER,-(R9)                      ;ASSUME SUPERVISOR MODE ACMODE
                 79     D4    058C              CLRL    -(R9)                                   ;ASSUEM /NOQUOTA
    79   01000000 8F    D0    058E              MOVL    #LNM$M_CREATE_IF,-(R9)                  ;ASSUME /NOATTR
       51    FAAB CF    9E    0595              MOVAB   LNM$PROCESS_DIRECTORY,R1                ;ASSUME /NOPARENT
            50   81     9A    059A              MOVZBL  (R1)+,R0                                ;
            79   50     7D    059D              MOVQ    R0,-(R9)                                ;SAVE THE DESCRIPTOR
                              05A0
              FA5D'     30    05A0      10$:    BSBW    DCL$GETDVAL                             ;GET NEXT DESCRIPTOR VALUE
            55   03     91    05A3      20$:    CMPB    #PTR_K_PARAMETR,R5                      ;ITEM TYPE PARAMETER?
                 03     12    05A6              BNEQ    30$                                     ;NO, THEN PROCESS QUALIFIER
               0124     31    05A8              BRW     200$                                    ;YES, THEN DONE
                              05AB
              FA52'     30    05AB      30$:    BSBW    DCL$GETNVAL                             ;GET QUALIFIER NUMBER
  00000000'8F    51     D1    05AE              CMPL    R1,#CLI$K_CRET_USER                     ;QUALIFIER MATCH?
                 45     13    05B5              BEQL    100$                                    ;YES, THEN BRANCH
  00000000'8F    51     D1    05B7              CMPL    R1,#CLI$K_CRET_SUPE                     ;QUALIFIER MATCH?
                 42     13    05BE              BEQL    110$                                    ;YES, THEN BRANCH
  00000000'8F    51     D1    05C0              CMPL    R1,#CLI$K_CRET_EXEC                     ;QUALIFIER MATCH?
                 3F     13    05C7              BEQL    120$                                    ;YES, THEN BRANCH
  00000000'8F    51     D1    05C9              CMPL    R1,#CLI$K_CRET_QUOT                     ;QUALIFIER MATCH?
                 3C     13    05D0              BEQL    130$                                    ;YES, THEN BRANCH
  00000000'8F    51     D1    05D2              CMPL    R1,#CLI$K_CRET_ATTR                     ;QUALIFIER MATCH?
                 4F     13    05D9              BEQL    140$                                    ;YES, THEN BRANCH
  00000000'8F    51     D1    05DB              CMPL    R1,#CLI$K_CRET_PARE                     ;QUALIFIER MATCH?
                 7B     13    05E2              BEQL    150$                                    ;YES, THEN BRANCH
  00000000'8F    51     D1    05E4              CMPL    R1,#CLI$K_CRET_PROT                     ;QUALIFIER MATCH?
                 03     12    05EB              BNEQ    40$                                     ;NO
               0085     31    05ED              BRW     160$                                    ;YES, THEN BRANCH
  00000000'8F    51     D1    05F0      40$:    CMPL    R1,#CLI$K_CRET_LOG                      ;QUALIFIER MATCH
                 A7     12    05F7              BNEQ    10$                                     ;NO, GET NEXT TOKEN
               00B7     31    05F9              BRW     175$                                    ;YES, THEN BRANCH
                              05FC
         10 A9   03     D0    05FC      100$:   MOVL    #PSL$C_USER,16(R9)                      ;SET USER MODE
```

```
                      9E  11  0600        BRB     10$                             ;GET NEXT TOKEN
             10 A9    02  D0  0602 110$:  MOVL    #PSL$C_SUPER,16(R9)             ;SET SUPER MODE
                      98  11  0606        BRB     10$                             ;GET NEXT TOKEN
             10 A9    01  D0  0608 120$:  MOVL    #PSL$C_EXEC,16(R9)              ;SET EXEC MODE
                      92  11  060C        BRB     10$                             ;GET NEXT TOKEN
                              060E
                 0C A9    D4  060E 130$:  CLRL    12(R9)                          ;ASSUME /NOQUOTA
                 10 53    E8  0611        BLBS    R3,131$                         ;BRANCH IF SO
                  F9E9'   30  0614        BSBW    DCL$GETDVAL                     ;GET QUOTA VALUE
               52   51    7D  0617        MOVQ    R1,R2                           ;COPY DESCRIPTOR
               51   01    D0  061A        MOVL    #1,R1                           ;SET DECIMAL RADIX
                  F9E0'   30  061D        BSBW    DCL$CNVNOEDIT                   ;CONVERT NUMBER TO BINARY
             0C A9   51   D0  0620        MOVL    R1,12(R9)                       ;SAVE THE VALUE AWAY
                   FF79   31  0624 131$:  BRW     10$                             ;GET NEXT TOKEN
                   FF79   31  0627 132$:  BRW     20$                             ;PROCESS NEXT TOKEN
                              062A
     08 A9   01000000 8F  D0  062A 140$:  MOVL    #LNM$M_CREATE_IF,8(R9)          ;ASSUME /NOATTRIBUTES
                  EF 53   E8  0632        BLBS    R3,131$                         ;BRANCH IF SO
                  F9C8'   30  0635 142$:  BSBW    DCL$GETDVAL                     ;GET ATTRIBUTE KEYWORD
               02   55    91  0638        CMPB    R5,#PTR_K_QUALVALU              ;SKIP IF NOT A QUALIFIER VALUE
                    EA    12  063B        BNEQ    132$                            ;
             4E 8F  62    91  063D        CMPB    (R2),#^A/N/                     ;NO_ALIAS KEYWORD?
                    06    12  0641        BNEQ    144$                            ;NO, THEN BRANCH
             08 A9  01    C8  0643        BISL    #LNM$M_NO_ALIAS,8(R9)           ;SET THE ATTRIBUTE
                    EC    11  0647        BRB     142$                            ;GET NEXT VALUE
             43 8F  62    91  0649 144$:  CMPB    (R2),#^A/C/                     ;CONFINE KEYWORD?
                    06    12  064D        BNEQ    146$                            ;NO, THEN BRANCH
             08 A9  02    C8  064F        BISL    #  [illegible]M$M_CONFINE,8(R9)  ;SET THE ATTRIBUTE
                    E0    11  0653        BRB     14[illegible]$                   ;GET NEXT VALUE
     08 A9   01000000 8F  CA  0655 146$:  BICL    #LNM$M_CREATE_IF,8(R9)          ;CLEAR THE ATTRIBUTE
                    D6    11  065D        BRB     142$                            ;GET NEXT VALUE
                              065F
          51     F9E1 CF  9E  065F 150$:  MOVAB   LNM$PROCESS_DIRECTORY,R1        ;ASSUME /NOPARENT
               50   81    9A  0664        MOVZBL  (R1)+,R0                        ;
               69   50    7D  0667        MOVQ    R0,(R9)                         ;SA[illegible] THE DESCRIPTOR
                 B7 53    E8  066A        BLBS    R3,131$                         ;BRANCH IF SO
                  F990'   30  066D        BSBW    DCL$GETDVAL                     ;GET TABLE NAME
               69   51    7D  0670        MOVQ    R[illegible],(R9)                ;SAVE THE DESCRIPTOR
                    AF    11  0673        BRB     131$                            ;GET NEXT TOKEN
                              0675 ;
                              0675 ;      SET LOGICAL NAME TABLE PROTECTION CODE
                              0675 ;
                  F988'   30  0675 160$:  BSBW    DCL$GETDVAL                     ;GET NEXT DESCRIPTOR VALUES
               55   02    91  0678        CMPB    #PTR_K_QUALVALU,R5              ;QUALIFIER VALUE?
                    AA    12  067B        BNEQ    132$                            ;NO, ALL DONE WITH PROTECTION.
                              067D
    FA07 CF    04   62    3A  067D        LOCC    (R2),#4,CLASS                   ;LOCATE PROTECTION CLASS
                    3A    13  0683        BEQL    180$                            ;IF EQL INVALID CLASS
                              0685
                    50    D7  0685        DECL    R0                              ;CALCULATE STARTING BIT NUMBER
          58   50   04    C5  0687        MULL3   #4,R0,R8
    14 A9   04   58   0F  F0  068B        INSV    #^XF,R8,#4,20(R9)               ;ASSUME NO ACCESS
               54   02    91  0691        CMPB    #PTR_K_COLON,R4                 ;PROTECTION VALUE SPECIFIED?
                    DF    12  0694        BNEQ    160$                            ;NO, TRY TO GET NEXT CLASS
                              0696
                  F967'   30  0696        BSBW    DCL$GETDVAL                     ;GET PROTECTION VALUE DESCRIPTOR
               57   51    D0  0699        MOVL    R1,R7                           ;SAVE LENGTH OF VALUE STRING
```

```
                                 069C
     F9E4 CF    04    82    3A   069C  165$:   LOCC    (R2)+,#4,ACCESS          ;LOCATE PROTECTION CODE
                      23    13   06A2          BEQL    185$                     ;IF EQUL INVALID PROTECTION CODE
                      50    D7   06A4          DECL    R0                       ;CALCULATE RELATIVE BIT NUMBER
                50    58    C0   06A6          ADDL    R8,R0                    ;CALCULATE ACTUAL BIT NUMBER
          00 14 A9    50    E5   06A9          BBCC    R0,20(R9),170$           ;ALLOW SPECIFIED ACCESS
                  EB  57    F5   06AE  170$:   SOBGTR  R7,165$                  ;ANY MORE TO SCAN?
                      C2    11   06B1          BRB     160$                     ;NO, TRY TO GET NEXT CLASS
                                 06B3        ;
                                 06B3        ;         PROCESS /LOG QUALIFIER
                                 06B3        ;
                   18 A9    94   06B3  175$:   CLRB    24(R9)                   ;ASSUME /LOG
                   03 53    E9   06B6          BLBC    R3,176$                  ;IT IS /LOG. FLAG OK AS IS
                   18 A9    96   06B9          INCB    24(R9)                   ;IT IS /NOLOG. SET FLAG
                    FF65    31   06BC  176$:   BRW     131$                     ;GET NEXT TOKEN
                                 06BF
                                 06BF  180$:   STATUS  IVKEYW                   ;SET INVALID KEYWORD
                            05   06C6          RSB                              ;EXIT
                                 06C7  185$:   STATUS  IVPROT                   ;SET INVALID PROTECTION CODE
                            05   06CE          RSB                              ;EXIT
                                 06CF
                79    51    7D   06CF  200$:   MOVQ    R1,-(R9)                 ;SAVE THE LOGICAL NAME DESR
                                 06D2
                                 06D2          $CRELNT_S  ATTR=16(R9),-         ;CREATE THE TABLE
                                 06D2                  QUOTA=20(R9),-           ;
                                 06D2                  TABNAM=(R9),-            ;
                                 06D2                  PARTAB=8(R9),-           ;
                                 06D2                  ACMODE=24(R9),-          ;
                                 06D2                  PROMSK=28(R9)            ;
                                 06EE
                                 06EE        ;
                                 06EE        ;         OUTPUT INFORMATION MESSAGES ABOUT THE TABLE CREATION
                                 06EE        ;
             40 20 A9       E8   06EE          BLBS    32(R9),280$              ;SKIP IF /NOLOG
                50    01    B1   06F2          CMPW    #SS$_NORMAL,R0           ;EXISTING TABLE NOT SUPER.?
                      09    12   06F5          BNEQ    210$                     ;NO, CHECK OTHER STATUS
     50    0003DE0B 8F      D0   06F7          MOVL    #CLI$_TABEXIST,R0        ;YES, TELL USER
                      28    11   06FE          BRB     270$                     ;
                                 0700
       50    0631 8F        B1   0700  210$:   CMPW    #SS$_SUPERSEDE,R0        ;EXISTING TABLE SUPERSEDED?
                      09    12   0705          BNEQ    220$                     ;NO, CHECK OTHER STATUS
     50    0003DE13 8F      D0   0707          MOVL    #CLI$_TABSUPER,R0        ;YES, TELL USER
                      18    11   070E          BRB     270$                     ;
                                 0710
       50    06B1 8F        B1   0710  220$:   CMPW    #SS$_LNMCREATED,R0       ;NEW TABLE CREATED?
                      1B    12   0715          BNEQ    280$                     ;NO, CHECK FOR CREATION ERROR
     50    0003DE1B 8F      DC   0717          MOVL    #CLI$_TABNOTFND,R0       ;ASSUME /SUPERSEDE SPECIFIED
  10 A9    01000000 8F      D3   071E          BITL    #LNM$M_CREATE_IF,16(R9)  ;WAS /SUPERSEDE SPECIFIED?
                      0D    12   0726          BNEQ    285$                     ;IF NOT, SKIP MESSAGE
                                 0728
                      69    9F   0728  270$:   PUSHAB  (R9)                     ;GET TABLE NAME DESCIPTOR
                51    01    D0   072A          MOVL    #1,R1                    ;SET FAO COUNT
                   F8D0'    30   072D          BSBW    DCL$FORMMSG              ;OUTPUT MESSAGE
                      03    11   0730          BRB     285$                     ;EXIT WITH STATUS NORMAL
                                 0732
                   07 50    E9   0732  280$:   BLBC    R0,290$                  ;BRANCH IF ERROR
                                 0735  285$:   STATUS  NORMAL                   ;RETURN SUCCESS
```

```
                                  05  073C    943 290$:    RSB                                  ;
```

```
                         073D         .SBTTL  SHOW LOGICAL NAME EQUIVALENCES
                         073D   ;+
                         073D   ; DCL$SHOWTRAN - SHOW LOGICAL NAME TRANSLATION
                         073D   ;
                         073D   ; THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SHOW LOGICAL
                         073D   ; NAME EQUIVALENCES DCLS COMMAND.
                         073D   ;
                         073D   ; INPUTS:
                         073D   ;
                         073D   ;       R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
                         073D   ;       R9 = ADDRESS OF SCRATCH STACK.
                         073D   ;       R10 = BASE ADDRESS OF COMMAND WORK AREA.
                         073D   ;       R11 = BASE ADDRESS OF PROCESS WORK AREA.
                         073D   ;
                         073D   ; OUTPUTS:
                         073D   ;
                         073D   ;       THE SPECIFIED LOGICAL NAME EQUIVALENCE FROM THE PROCESS
                         073D   ;       LOGICAL NAME TABLE IS WRITTEN TO THE OUTPUT STREAM.
                         073D   ;-
                         073D   DCL$SHOWTRAN::                                          ;SHOW THE TRANSLATION FOR A NAME
                         073D
                         073D   ;++
                         073D   ; Stack layout:
                         073D   ;
                         073D   ;       ;---------------;
                         073D   ;       ; Table name    ;
                         073D   ;       ;  descriptor   ;
                         073D   ;       ;---------------;
                         073D   ;       ; Logical name  ;
                         073D   ;       ;  descriptor   ;
                         073D   ;       ;---------------;
                         073D   ;       ; Equival name  ;
                         073D   ;       ;  descriptor   ;
                         073D   ;       ;---------------;
                         073D   ;       ; Item list ... ;
                         073D   ;       ;               ;
                         073D   ;
                         073D   ;--
                         073D   ;
                         073D   ; Parse the command string.
                         073D   ;
51     F8F3 CF     9E    073D          MOVAB   LNM$DCL_LOGICAL,R1              ;SET DEFAULT LOGICAL NAME TABLE
     50     81     9A    0742          MOVZBL  (R1)+,R0                        ;
     7E     50     7D    0745          MOVQ    R0,-(SP)                        ;
                         0748
       F8B5'       30    0748   10$:   BSBW    DCL$GETDVAL                     ;GET FIRST TOKEN
     55     03     91    074B          CMPB    #PTR_K_PARAMETR,R5              ;IS IT A PARAMETER
            06     12    074E          BNEQ    15$                             ;NO, THEN PROCESS /TABLE
                         0750          ASSUME  PTR_V_KEYWORD EQ 21
  F4 53     01     E0    0750          BBS     #1,R3,10$                       ;IGNORE OPTION KEYWORD
            16     11    0754          BRB     20$                             ;PROCESS THE LOGICAL NAME
                         0756
51     F8DA CF     9E    0756   15$:   MOVAB   LNM$DCL_LOGICAL,R1              ;ASSUME /NOTABLE
     50     81     9A    075B          MOVZBL  (R1)+,R0                        ;
     6E     50     7D    075E          MOVQ    R0,(SP)                         ;
      E4 53        E8    0761          BLBS    R3,10$                          ;BRANCH IF SO
       F899'       30    0764          BSBW    DCL$GETDVAL                     ;GET TABLE NAME
```

```
          6E      51   7D  0767              MOVQ    R1,(SP)                          ;SAVE IT
                  DC   11  076A              BRB     10$                              ;GET NEXT TOKEN
                           076C
          7E      51   7D  076C      20$:    MOVQ    R1,-(SP)                         ;SAVE LOGICAL NAME DESCR
                           076F
                           076F      ;
                           076F      ; Create item list and perform translation.
                           076F      ;
                      7E   D4  076F          CLRL    -(SP)                            ;BUILD ITEM LIST
       7E    F4   AE   9E  0771              MOVAB   -12(SP),-(SP)                    ;SET ADDR OF RETURN LENGTH
     7E    F896   CA   9E  0775              MOVAB   WRK_G_INPBUF(R10),-(SP)          ;BUILD TABLE NAME DESCRIPTOR
  7E   00040100   8F   D0  077A              MOVL    #LNM$_TABLE@16+WRK_C_INPBUFSIZ,-(SP) ;SET ITEM TYPE AND LENGTH
       7E    F4   AE   9E  0781              MOVAB   -12(SP),-(SP)                    ;SET ADDR OF RETURN LENGTH
     50      51   06   C1  0785              ADDL3   #6,R1,R0                         ;LENGTH OF RESULT BEFORE EQUIV
  7E    04   A8   50   C1  0789              ADDL3   R0,4(R8),-(SP)                   ;BUILD EQUIV NAME DESCRIPTOR
     7E      68   50   C3  078E              SUBL3   R0,(R8),-(SP)                    ;
       02    AE   02   B0  0792              MOVW    #LNM$_STRING,2(SP)               ;SET ITEM TYPE
             57   5E   D0  0796              MOVL    SP,R7                            ;SET ITEM LIST ADDR
                           0799
                           0799              $TRNLNM_S  TABNAM=36(R7),-               ;TRANSLATE THE LOGICAL NAME
                           0799                         LOGNAM=28(R7),-               ;
                           0799                         ITMLST=(R7)                   ;
                           07AC
             01   50   D1  07AC              CMPL    R0,#SS$_NORMAL                   ;TEST FOR SUCCESSFUL TRANSLATION
                  0E   13  07AF              BEQL    30$                              ;BRANCH IF SUCCESS
                  67   B4  07B1              CLRW    (R7)                             ;ELSE CLEAR BYTE COUNT OF RESULTANT
     51    F8B0   CF   9E  07B3              MOVAB   UNDEFINED,R1                     ;INDICATE UNDEFINED
             50   81   9A  07B8              MOVZBL  (R1)+,R0                         ;
       0C    A7   50   7D  07BB              MOVQ    R0,12(R7)                        ;
                           07BF
                           07BF      ;
                           07BF      ; Strip off escape sequences.
                           07BF      ;
          02 A7        B4  07BF      30$:    CLRW    2(R7)                            ;CLEAR ITEM TYPE
          52      67   9E  07C2              MOVAB   (R7),R2                          ;GET ADDRESS OF EQUIV DESCRIPTOR
                  62   D5  07C5              TSTL    (R2)                             ;ZERO LENGTH EQUIV?
                  11   13  07C7              BEQL    40$                              ;IF EQL YES
       04    B2   1B   91  07C9              CMPB    #27,@4(R2)                       ;FIRST CHARACTER ESCAPE?
                  0B   12  07CD              BNEQ    40$                              ;IF NEQ NO
       04    A2   04   C0  07CF              ADDL    #4,4(R2)                         ;POINT PAST EQUIV HEADER
             62   04   C2  07D3              SUBL    #4,(R2)                          ;REDUCE LENGTH OF EQUIV BY HEADER
                  02   18  07D6              BGEQ    40$                              ;IF GEQ OKAY
                  62   D4  07D8              CLRL    (R2)                             ;CLEAR EQUIV LENGTH
                           07DA
                           07DA      ;
                           07DA      ; Output the message.
                           07DA      ;
     51    F893   CF   9E  07DA      40$:    MOVAB   LOGICALMSG,R1                    ;GET ADDRESS OF ASCIC FAO STRING
             50   81   9A  07DF              MOVZBL  (R1)+,R0                         ;MAKE INTO DESCRIPTOR
             7E   50   7D  07E2              MOVQ    R0,-(SP)                         ;PUSH ONTO STACK
             50   5E   D0  07E5              MOVL    SP,R0                            ;GET DESCRIPTOR ADDRESSES
       51    1C   A7   7E  07E8              MOVAQ   28(R7),R1                        ;
          52      67   7E  07EC              MOVAQ   (R7),R2                          ;
       53    0C   A7   7E  07EF              MOVAQ   12(R7),R3                        ;
                           07F3
                           07F3              $FAO_S  (R0),(R8),(R8),R1,R2,R3          ;FORMAT OUTPUT MESSAGE
                           0806
```

```
           5E   2C A7   9E  0806  1059          MOVAB   44(R7),SP                       ;RESTORE THE STACK
              51   68   7D  080A  1060          MOVQ    (R8),R1                         ;GET OUTPUT MESSAGE PARAMETERS
                 F7F0'  30  080D  1061          BSBW    DCL$MSGOUT                      ;OUTPUT MESSAGE
                            0810  1062          STATUS  NORMAL                          ;RETURN SUCCESS
                        05  0817  1063          RSB
                            0818  1064
                            0818  1065          .END
```

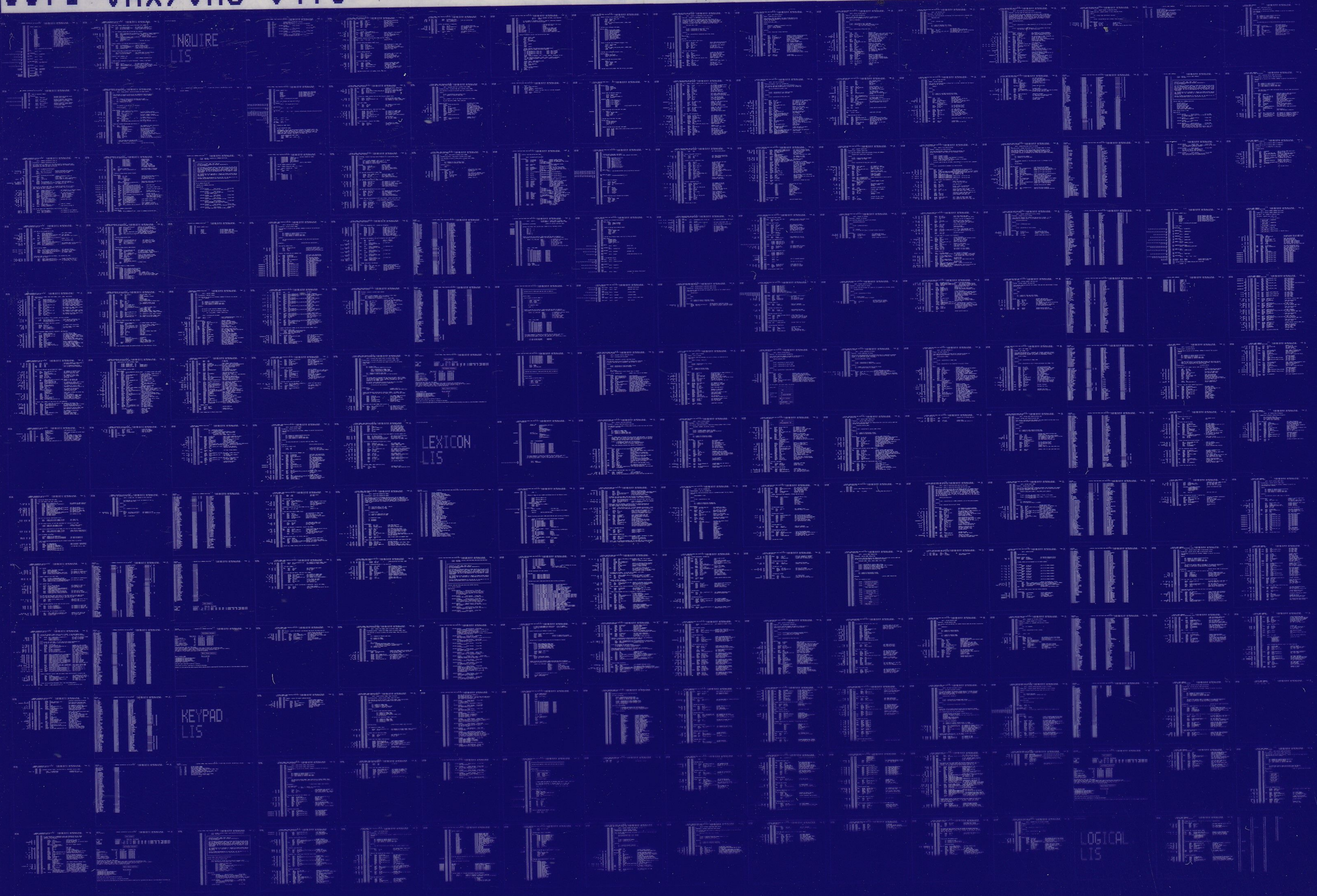
```
                                                  D  1
LOGICAL                          - LOGICAL NAME COMMANDS          16-SEP-1984 00:08:00  VAX/VMS Macro V04-00        Page  27
Symbol table                                                       4-SEP-1984 23:41:57  [DCL.SRC]LOGICAL.MAR;1           (13)

$$T1                      = 00000000                  EQUNAM                        = 00000020
$$T2                      = 00000006                  GET_TRAN_ATTR                   00000544 R       02
ACCESS                      00000086 R       02       LNM$DCL_LOGICAL                 00000034 R       02
ACMODE                    = 0000000C                  LNM$FILE_DEV                    0000005A R       02
ATTR_M                    = 00000002                  LNM$GROUP                       0000001F R       02
ATTR_V                    = 00000001                  LNM$JOB                         00000017 R       02
CLASS                       0000008A R       02       LNM$M_CONCEALED               = 00000100
CLI$K_ALLO_GENE             *********    X   02       LNM$M_CONFINE                 = 00000002
CLI$K_CRET_ATTR             *********    X   02       LNM$M_CREATE_IF               = 01000000
CLI$K_CRET_EXEC             *********    X   02       LNM$M_NO_ALIAS                = 00000001
CLI$K_CRET_LOG              *********    X   02       LNM$M_TERMINAL                = 00000200
CLI$K_CRET_PARE             *********    X   02       LNM$PROCESS                     0000000B R       02
CLI$K_CRET_PROT             *********    X   02       LNM$PROCESS_DIRECTORY           00000044 R       02
CLI$K_CRET_QUOT             *********    X   02       LNM$SYSTEM                      00000029 R       02
CLI$K_CRET_SUPE             *********    X   02       LNM$_ATTRIBUTES               = 00000003
CLI$K_CRET_USER             *********    X   02       LNM$_STRING                   = 00000002
CLI$K_DEFI_EXEC             *********    X   02       LNM$_TABLE                    = 00000004
CLI$K_DEFI_GROU             *********    X   02       LOGICALMSG                      00000071 R       02
CLI$K_DEFI_JOB              *********    X   02       LOGNAM                        = 00000018
CLI$K_DEFI_LOG              *********    X   02       LOG_M                         = 00000001
CLI$K_DEFI_NAME             *********    X   02       LOG_V                         = 00000000
CLI$K_DEFI_PROC             *********    X   02       NAME_ATTR                     = 00000004
CLI$K_DEFI_SUPE             *********    X   02       OUTPUTNAM                       00000000 R       02
CLI$K_DEFI_SYST             *********    X   02       PRC_B_CONTINUE                  000000F3
CLI$K_DEFI_TABL             *********    X   02       PRC_B_DEFRADIX                  000000AE
CLI$K_DEFI_TRAN             *********    X   02       PRC_B_EXMDEPMOD                 000000AD
CLI$K_DEFI_USER             *********    X   02       PRC_B_EXMDEPWID                 000000AC
CLI$_ALLOC                = 0003DDE3                  PRC_B_EXONLYL                   0000012D
CLI$_IVKEYW               = 00038060                  PRC_B_FLAGS2                    000000AF
CLI$_IVPROT               = 00038070                  PRC_B_IMGFLAG                   0C000078
CLI$_NORMAL               = 00030001                  PRC_B_OUTFLAGS                  0000012C
CLI$_SUPERSEDE            = 0003DDEB                  PRC_B_PROMPTLEN                 000000F0
CLI$_TABEXIST             = 0003DE0B                  PRC_C_LENGTH                    00000534
CLI$_TABNOTFND            = 0003DE1B                  PRC_G_COMMANDS                  00000133
CLI$_TABSUPER             = 0003DE13                  PRC_G_PROMPT                    000000F4
COMMON_CRELNM               000002AD R       02       PRC_K_LENGTH                    00000534
COMMON_QUAL                 000003C7 R       02       PRC_L_CURRKEY                   00000048
DCL$ALLOCATE                0000008E RG      02       PRC_L_EXMDEPADR                 000000A8
DCL$ASSIGN                  000001A3 RG      02       PRC_L_EXTARG                    00000094
DCL$CNVNOEDIT               *********    X   02       PRC_L_EXTBLK                    0000008C
DCL$COMPRESS                *********    X   02       PRC_L_EXTCOD                    0000009C
DCL$COMPSTRING              *********    X   02       PRC_L_EXTHND                    00000090
DCL$CREATE_OUTPUT           *********    X   02       PRC_L_EXTPRM                    00000098
DCL$CRETABLE                00000582 RG      02       PRC_L_IDFLNK                    000000BC
DCL$DEALLOCAT               C0000314 RG      02       PRC_L_IMGACTSTS                 00000080
DCL$DEASSIGN                0000033B RG      02       PRC_L_INDCLOCK                  0000007C
DCL$DEFINE                  00000231 RG      02       PRC_L_INDEPTH                   0000005C
DCL$FORMMSG                 *********    X   02       PRC_L_INDFAB                    0000001C
DCL$GETDVAL                 *********    X   02       PRC_L_INDINPRAB                 00000014
DCL$GETNVAL                 *********    X   02       PRC_L_INDOUTRAB                 00000018
DCL$MSGOUT                  *********    X   02       PRC_L_INPRAB                    00000008
DCL$OPEN_OUTPUT             *********    X   02       PRC_L_LASTKEY                   0000004C
DCL$RESTORE_OUTPUT          *********    X   02       PRC_L_LSTSTATUS                 000000B0
DCL$SHOWTRAN                0000073D RG      02       PRC_L_ONCTLY                    000000B8
DEF_M                     = 00000004                  PRC_L_ONERROR                   0000006C
DEF_PROT                  = 0000FF00                  PRC_L_OUTOFBAND                 000000B4
DEF_V                     = 00000002                  PRC_L_OUTRAB                    0000000C
```

```
PRC_L_OUTRABCTX              00000118          PTR_V_KEYWORD              = 00000015
PRC_L_PPFLIST                00000070          PTR_V_NEGATE               = 00000014
PRC_L_RECALLPTR              0000012F          QUAL                       = 00000008
PRC_L_RESTART                00000058          SS$_LNMCREATED             = 000006B1
PRC_L_SAVAP                  00000000          SS$_NORMAL                 = 00000001
PRC_L_SAVFP                  00000004          SS$_NOSUCHDEV              = 00000908
PRC_L_SEVERITY               00000050          SS$_SUPERSEDE              = 00000631
PRC_L_SPWN                   000000C0          SYS$ALLOC                    ******** GX   02
PRC_L_STACKLM                000000A4          SYS$CRELNM                   ******** GX   02
PRC_L_STACKPT                000000A0          SYS$CRELNT                   ******** GX   02
PRC_L_STATUS                 00000054          SYS$DALLOC                   ******** GX   02
PRC_L_STS                    00000084          SYS$DELLNM                   ******** GX   02
PRC_L_STV                    00000088          SYS$FAO                      ********  X   02
PRC_L_SYMBOL                 00000060          SYS$TRNLNM                   ******** GX   02
PRC_L_TMBX                   00000074          TABNAM                     = 00000010
PRC_L_TRMLIST                00000010          TESTOUT                      00000391 R    02
PRC_Q_ALLOCREG               00000020          TRAN_ATTR                  = 00000000
PRC_Q_COMMAND                000000E0          UNDEFINED                    00000067 R    02
PRC_Q_FLUSHTIME              000000D0          WRK_B_CMDOPT                 FFFFFFC3
PRC_Q_GLOBAL                 00000028          WRK_B_MAXPARM                FFFFFFD0
PRC_Q_IMAGENAME              000000D8          WRK_B_MINPARM                FFFFFFD1
PRC_Q_KEYPAD                 00000040          WRK_B_PARMCNT                FFFFFFCE
PRC_Q_LABEL                  00000030          WRK_B_PARMSUM                FFFFFFCF
PRC_Q_LOCAL                  00000038          WRK_B_RECALLCNT              FFFFFFC5
PRC_Q_SAVEPRIV               000000E8          WRK_B_VALLEV                 FFFFFFC4
PRC_T_OUTDVI                 0000011C          WRK_B_VERBTYP                FFFFFFC2
PRC_W_ASTIOSB                000000C6          WRK_C_INPBUFSIZ            = 00000100
PRC_W_ASTRETN                000000C8          WRK_C_LENGTH                 FFFFF486
PRC_W_ASTSTATUS              000000C4          WRK_G_BUFFER                 FFFFF492
PRC_W_ATTMBX                 0000007A          WRK_G_INPBUF                 FFFFF896
PRC_W_FLAGS                  00000068          WRK_G_RESULT                 FFFFF9B6
PRC_W_INPCHAN                00000064          WRK_K_LENGTH                 FFFFF486
PRC_W_ONLEVEL                0000006A          WRK_L_CHARPTR                FFFFF48E
PRC_W_OUTIFI                 00000114          WRK_L_DISALLOW               FFFFFFE6
PRC_W_OUTISI                 00000116          WRK_L_ERRORRTN               FFFFF9AE
PRC_W_OUTMBXCHN              000000CA          WRK_L_EXPANDPTR              FFFFF486
PRC_W_OUTMBXREF              000000CE          WRK_L_IMAGE                  FFFFFFE2
PRC_W_OUTMBXSIZ              000000CC          WRK_L_MARKPTR                FFFFF48A
PRC_W_PMPTCTRL               000000F1          WRK_L_PAROUT                 FFFFFFD2
PRC_W_WAITIOSB               00000066          WRK_L_PMPTADDR               FFFFF9A2
PSL$C_EXEC                 = 00000001          WRK_L_PROMPTRTN              FFFFF9A6
PSL$C_SUPER                = 00000002          WRK_L_PROPTR                 FFFFFFC6
PSL$C_USER                 = 00000003          WRK_L_QUABLK                 FFFFFFCA
PTR_B_LEVEL                  00000004          WRK_L_READRTN                FFFFF9AA
PTR_B_NUMBER                 00000005          WRK_L_RECALLPTR              FFFFFFEA
PTR_B_PARMCNT                00000006          WRK_L_RSLEND                 FFFFFFB6
PTR_B_VALUE                  00000000          WRK_L_RSLNXT                 FFFFFFBA
PTR_C_LENGTH                 0000000C          WRK_L_SAVAP                  FFFFFFF8
PTR_K_COLON                = 00000002          WRK_L_SAVFP                  FFFFFFFC
PTR_K_COMMA                = 00000005          WRK_L_SAVSP                  FFFFFFF4
PTR_K_ENDLINE              = 00000004          WRK_L_SIGNALRTN              FFFFFFD6
PTR_K_LENGTH                 0000000C          WRK_L_SPECRTN                FFFFF9B2
PTR_K_PARAMETR             = 00000003          WRK_L_TAB_VEC                FFFFFFDE
PTR_K_QUALVALU             = 00000002          WRK_L_VERB                   FFFFFFBE
PTR_L_DESCR                  00000000          WRK_W_FLAGS                  FFFFFFF0
PTR_L_ENTITY                 00000008          WRK_W_FLAGS2                 FFFFFFF2
PTR_V_FLAGS                = 00000014          WRK_W_IMGCHAN                FFFFFFEE
```

```
LOGICAL                          - LOGICAL NAME COMMANDS                  16-SEP-1984 00:08:00  VAX/VMS Macro V04-00        Page  29
Symbol table                                                               4-SEP-1984 23:41:57  [DCL.SRC]LOGICAL.MAR;1            (13)

WRK_W_PMPTLEN                     FFFFF99E

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

PSECT name                       Allocation           PSECT No.  Attributes
----------                       ----------           ---------  ----------
. ABS  .                         00000000  (     0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                            FFFFFFFC  (     0.)  01 (   1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
DCL$ZCODE                        00000818  ( 2072.)  02 (   2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

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

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                    15     00:00:00.08     00:00:01.26
Command processing               101     00:00:00.65     00:00:05.99
Pass 1                           331     00:00:13.05     00:00:33.89
Symbol table sort                  0     00:00:01.69     00:00:04.36
Pass 2                           183     00:00:03.27     00:00:09.90
Symbol table output               27     00:00:00.18     00:00:00.64
Psect synopsis output              2     00:00:00.03     00:00:00.05
Cross-reference output             0     00:00:00.00     00:00:00.00
Assembler run totals             659     00:00:18.96     00:00:56.11

The working set limit was 1500 pages.
69087 bytes (135 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 980 non-local and 104 local symbols.
1065 source lines were read in Pass 1, producing 20 object records in Pass 2.
43 pages of virtual memory were used to define 28 macros.

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

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[SYSLIB]SYSBLDMLB.MLB;1                    0
_$255$DUA28:[DCL.OBJ]DCL.MLB;1                         6
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     16
TOTALS (all libraries)                                22

1132 GETS were required to define 22 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:LOGICAL/OBJ=OBJ$:LOGICAL MSRC$:LOGICAL/UPDATE=(ENH$:LOGICAL)+EXECML$/LIB+LIB$:DCL/LIB+SYS$LIBRARY:SYSBLDMLB/LIB
```

0071 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


INQUIRE
LIS

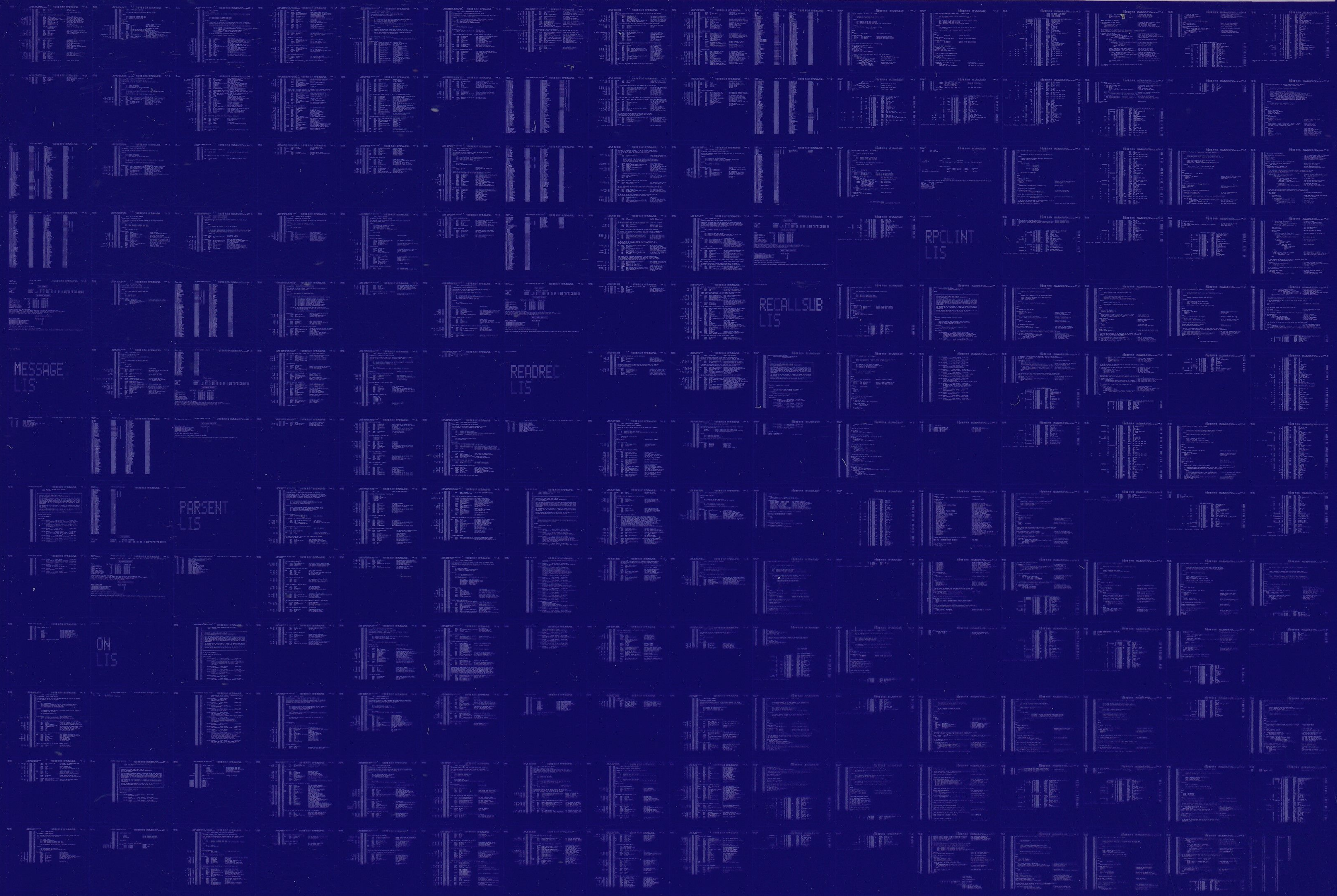
LEXICON
LIS

KEYPAD
LIS

LOGICAL
LIS

0072 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


RPCLINT
LIS

RECALLSUB
LIS

MESSAGE
LIS

READREC
LIS

PARSENT
LIS

ON
LIS