```
DDDDDDDDDDD         CCCCCCCCCCC  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
DDDDDDDDDDDD        CCCCCCCCCCCC LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC LLLLLLLLLLLLLLL
```

FILEID**DISALLOW

```
DDDDDDDD      IIIIII      SSSSSSSS    AAAAAA    LL          LL          OOOOOO    WW      WW
DDDDDDDD      IIIIII      SSSSSSSS    AAAAAA    LL          LL          OOOOOO    WW      WW
DD      DD      II      SS          AA    AA    LL          LL        OO    OO    WW      WW
DD      DD      II      SS          AA    AA    LL          LL        OO    OO    WW      WW
DD      DD      II      SS          AA    AA    LL          LL        OO    OO    WW      WW
DD      DD      II      SS          AA    AA    LL          LL        OO    OO    WW      WW
DD      DD      II        SSSSSS    AA    AA    LL          LL        OO    OO    WW      WW
DD      DD      II        SSSSSS    AA    AA    LL          LL        OO    OO    WW      WW
DD      DD      II              SS  AAAAAAAAAA  LL          LL        OO    OO    WW  WW  WW
DD      DD      II              SS  AAAAAAAAAA  LL          LL        OO    OO    WW  WW  WW
DD      DD      II              SS  AA    AA    LL          LL        OO    OO    WWWW  WWWW      ....
DD      DD      II              SS  AA    AA    LL          LL        OO    OO    WWWW  WWWW      ....
DDDDDDDD      IIIIII    SSSSSSSS    AA    AA    LLLLLLLLLL  LLLLLLLLLL    OOOOOO    WW      WW      ....
DDDDDDDD      IIIIII    SSSSSSSS    AA    AA    LLLLLLLLLL  LLLLLLLLLL    OOOOOO    WW      WW      ....

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

Table of contents

```
0000     1          .TITLE  DISALLOW - Evaluate Disallow Expressions
0000     2          .IDENT  'V04-000'
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28 ; AUTHOR:       Peter George 01-MAR-83
0000    29 ;
0000    30 ; ABSTRACT:     These routines are called to evaluate the disallow expression
0000    31 ;               for a command that has just been successfully parsed.
0000    32 ;
0000    33 ; MODIFICATIONS:
0000    34 ;
0000    35 ;       V03-006 HWS0098         Harold Schultz          02-Aug-1984
0000    36 ;               Treat all qualifiers as global for purposes of the
0000    37 ;               disallow expression. Disregard notion of a "local"
0000    38 ;               qualifier for the present. (the question of local
0000    39 ;               qualifiers will be addressed in a future version)
0000    40 ;
0000    41 ;       V03-005 HWS0089         Harold Schultz          22-Jul-1984
0000    42 ;               Fix checking for presence of local qualifiers.
0000    43 ;
0000    44 ;       V03-004 PCG0004         Peter George            03-May-1984
0000    45 ;               Allowing checking of keywords on negated qualifiers.
0000    46 ;
0000    47 ;       V03-003 PCG0003         Peter George            06-Dec-1983
0000    48 ;               Add NEG operator.
0000    49 ;
0000    50 ;       V03-002 PCG0002         Peter George            11-May-1983
0000    51 ;               Make path list search independent of token sorting.
0000    52 ;
0000    53 ;       V03-001 PCG0001         Peter George            30-Apr-1983
0000    54 ;               Make the expression evaluation more robust.
0000    55 ;---
0000    56
```

```
                     ;
                     ; Macro library calls
                     ;
                             $$CLITABDEF                     ; Define table structures
                             PRCDEF                          ; Define process work area
                             WRKDEF                          ; Define command work area
                             PTRDEF                          ; Define result parse descriptor
                     
00000000                     .PSECT  DCL$ZCODE,BYTE,RD,NOWRT
```

```
                        0000     68          .SBTTL  Evaluate Disallow Expression
                        0000     69 ;---
                        0000     70 ; DCL$EVAL_DISALLOW - Evaluate disallow expression
                        0000     71 ;
                        0000     72 ; This routine is called after the command verb has been parsed, to detect
                        0000     73 ; qualifier and parameter conflicts.
                        0000     74 ;
                        0000     75 ; INPUTS:
                        0000     76 ;
                        0000     77 ;       R10 =   Address of command work area (WRK)
                        0000     78 ;
                        0000     79 ; OUTPUTS:
                        0000     80 ;
                        0000     81 ;       R0 =    Value of disallow expression - true or false
                        0000     82 ;
                        0000     83 ; REGISTERS USED GLOBALLY IN EXPRESSION EVALUATION:
                        0000     84 ;
                        0000     85 ;       R1 =    Address of expression block
                        0000     86 ;       R7 =    Address of ENT block associated with last
                        0000     87 ;               conflicting entity
                        0000     88 ;       R8 =    Address of PTR block associated with last
                        0000     89 ;               conflicting entity
                        0000     90 ;       R9 =    The number of the parameter providing the qualifier context
                        0000     91 ;
                        0000     92 ;+++
                        0000     93
                        0000     94 DCL$EVAL_DISALLOW::
                        0000     95
                        0000     96 ;
                        0000     97 ; Check for conflicts on a per-parameter basis.
                        0000     98 ;
                        0000     99 ;       CLRL    R9                              ; Assume there are no parameters
                        0000    100 ;       TSTB    WRK_B_PARMSUM(R10)              ; Are there?
                        0000    101 ;       BEQL    10$                             ; No, then skip
                        0000    102 ;       INCL    R9                              ; Yes, then start with first
    59    CF AA     90  0000    103         MOVB    WRK_B_PARMSUM(R10),R9           ; Get total # of parameters
    51    E6 AA     D0  0004    104 10$:    MOVL    WRK_L_DISALLOW(R10),R1          ; Is there a disallow expression?
            06      13  0008    105         BEQL    95$                             ; No, then skip
          0057      30  000A    106         BSBW    DISPATCH                        ; Yes, evaluate it
          08 50     E8  000D    107         BLBS    R0,80$                          ; If success, then signal error
                        0010    108 ;       INCL    R9                              ; Set for next parameter
                        0010    109 ;       CMPB    R9,WRK_B_PARMSUM(R10)           ; Are there more parameters?
                        0010    110 ;       BLEQU   10$                             ; Yes, then continue
                        0010    111 95$:    STATUS  NORMAL                          ; Set normal return status
                  05    0017    112         RSB                                     ; Return
                        0018    113
                        0018    114 ;
                        0018    115 ; Set up the conflict error message.  Use actually command string if available.
                        0018    116 ; Otherwise, use "NO" + the entity name.
                        0018    117 ;
              58    D5  0018    118 80$:    TSTL    R8                              ; Was entity explicitly present?
              12    13  001A    119         BEQL    85$                             ; No, then use entity block
        08    00    EF  001C    120         EXTZV   #PTR_V_VALUE,#PTR_S_VALUE,-     ; Get length of name
        56    68        001F    121                 PTR_[_DESCR(R8),R6              ;
        0C    08    EF  0021    122         EXTZV   #PTR_V_OFFSET,#PTR_S_OFFSET,-   ; Get address of name
        57    68        0024    123                 PTR_[_DESCR(R8),R7              ;
57    F492 CA47   9E  0026    124         MOVAB   WRK_G_BUFFER(R10)[R7],R7        ;
```

```
                     23   11  002C   125          BRB     90$                                     ;
                                 002E   126
                58   57   D0  002E   127 85$:     MOVL    R7,R8                                   ; Get address of ENT block
             50   16 A8   32  0031   128          CVTWL   ENT_W_NAME(R8),R0                       ; Get offset to ascic name
              57   6840   9E  0035   129          MOVAB   (R8)[R0],R7                             ; Get address of ascic name
                56   87   9A  0039   130          MOVZBL  (R7)+,R6                                ; Get name length/address
F492 CA         4F4E 8F   B0  003C   131          MOVW    #^A/NO/,WRK_G_BUFFER(R10)               ; Preface the name with "NO"
F494 CA         67   56   28  0043   132          MOVC3   R6,(R7),WRK_G_BUFFER+2(R10)             ; Move string into the buffer
          57    F492 CA   9E  0049   133          MOVAB   WRK_G_BUFFER(R10),R7                    ; Set address of string
                56   02   C0  004E   134          ADDL    #2,R6                                   ; Add in length of "NO"
                                 0051   135
          F48A CA    57   D0  0051   136 90$:     MOVL    R7,WRK_L_MARKPTR(R10)                   ; Set start of string
F486 CA         57   56   C1  0056   137          ADDL3   R6,R7,WRK_L_EXPANDPTR(R10)              ; Set end of string
 50    00000000'8F       D0  005C   138          MOVL    #CLI$_CONFLICT,R0                       ; Set conflicting entity status
                         05  0063   139          RSB                                             ; Return
```

```
                     0064   141          .SBTTL  Dispatch to Disallow Expression Evaluator
                     0064   142 ;---
                     0064   143 ; DISPATCH - Dispatch to disallow expression evaluator
                     0064   144 ;
                     0064   145 ; This routine is called to invoke the appropriate routines for evaluating
                     0064   146 ; a disallow expression.
                     0064   147 ;
                     0064   148 ; INPUTS:
                     0064   149 ;
                     0064   150 ;       R1 =    Address of expression block
                     0064   151 ;       R10 =   Address of command work area (WRK)
                     0064   152 ;
                     0064   153 ; OUTPUTS:
                     0064   154 ;
                     0064   155 ;       R0 =    Value of disallow expression - true or false
                     0064   156 ;
                     0064   157 ;+++
                     0064   158
                     0064   159 DISPATCH:
                     0064   160
  03 A1    8F        0064   161          CASEB   EXP_B_SUBTYPE(R1),-                     ; Dispatch to action routine
07   01              0067   162                  #EXP_K_PATH,#EXP_K_NEG                  ;
        0011'        0069   163 10$:      .WORD   PATH-10$                                ; Action routines
        0105'        006B   164          .WORD   NOT-10$                                 ;
        0111'        006D   165          .WORD   ANY2-10$                                ;
        0139'        006F   166          .WORD   AND-10$                                 ;
        015A'        0071   167          .WORD   OR-10$                                  ;
        017A'        0073   168          .WORD   XOR-10$                                 ;
        01A5'        0075   169          .WORD   NEG-10$                                 ;
     50   D4        0077   170          CLRL    R0                                      ; Clear value if error
          05        0079   171          RSB                                             ; Return
                    007A   172
```

```
                     007A   174          .SBTTL  Evaluate Path Expression
                     007A   175 ;---
                     007A   176 ; PATH  - Evaluate path expression
                     007A   177 ;
                     007A   178 ; This routine is called to evaluate a path expression.
                     007A   179 ;
                     007A   180 ; INPUTS:
                     007A   181 ;
                     007A   182 ;       R1 =    Address of expression block
                     007A   183 ;       R9 =    Number of parameter to conduct local qualifier search at
                     007A   184 ;       R10 =   Address of command work area (WRK)
                     007A   185 ;
                     007A   186 ; OUTPUTS:
                     007A   187 ;
                     007A   188 ;       R0 =    Value of expression - true or false
                     007A   189 ;       R5,R6 are trashed
                     007A   190 ;       R7 =    Address of ENT block associated with last
                     007A   191 ;               conflicting entity
                     007A   192 ;       R8 =    Address of PTR block associated with last
                     007A   193 ;               conflicting entity
                     007A   194 ;       R9 =    The number of the parameter providing the qualifier context
                     007A   195 ;
                     007A   196 ;+++
                     007A   197
                     007A   198 ;
                     007A   199 ; Init state for the search.
                     007A   200 ;
        7E   52   7D 007A   201 PATH:   MOVQ    R2,-(SP)                                ; Save R2,R3
        52   01   D0 007D   202        MOVL    #1,R2                                   ; Init value level
     53   06   A1   3C 0080   203        MOVZWL  EXP_W_TRO_COUNT(R1),R3                  ; Get operand count
  58   F9AA   CA   9E 0084   204        MOVAB   WRK_G_RESULT-PTR_K_LENGTH(R10),R8       ; Get addr of PTR blocks
     55   04   A1   9E 0089   205        MOVAB   EXP_L_OPERAND_LIST-4(R1),R5             ; Get addr of entity block l
                     008D   206
                     008D   207 ;
                     008D   208 ; If the path specifies a parameter, find the first parameter value that
                     008D   209 ; we could possibly be a value of the specified parameter.
                     008D   210 ;
57   6542   DE   AA   C1 008D   211        ADDL3   WRK_L_TAB_VEC(R10),(R5)[R2],R7          ; Get entity block address
     03   A7   02   91 0093   212        CMPB    #ENT_K_QUALIFIER,ENT_B_SUBTYPE(R7)      ; Is it a qualifier?
                  2A   13 0097   213        BEQL    7$                                      ; Yes, then process it
     56   14   A7   9A 0099   214        MOVZBL  ENT_B_NUMBER(R7),R6                     ; Get parameter number
                     009D   215
        58   0C   C0 009D   216 5$:     ADDL    #PTR_K_LENGTH,R8                        ; Get next PTR block
  68   04   1C   ED 00A0   217        CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; End of line?
                  04   00A4   218                #PTR_K_ENDLINE                          ;
                  5C   13 00A5   219        BEQL    85$                             ; Yes, return false
  68   04   1C   ED 00A7   220        CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; Parameter?
                  03   00AB   221                #PTR_K_PARAMETR                         ;
                  EF   12 00AC   222        BNEQ    5$                              ; No, skip it
     56   06   A8   91 00AE   223        CMPB    PTR_B_PARMCNT(R8),R6                    ; Is is the one we want?
                  E9   12 00B2   224        BNEQ    5$                              ; No, get next PTR
                     00B4   225 ;
                     00B4   226 ; We are processing and have found a parameter specification.  If we are
                     00B4   227 ; not looking for a specific keyword, return success now.  If we are, then
                     00B4   228 ; set up the index and ptr block starting address for the search.
                     00B4   229 ;
        53   01   D1 00B4   230        CMPL    #1,R3                                   ; Only one entity address?
```

```
                     45  13  00B7   231          BEQL    80$                                     ; Yes, return true
                     53  D7  00B9   232          DECL    R3                                      ; Put in synch with level
                55   04  C0  00BB   233          ADDL    #4,R5                                   ; Shift the table address
                58   0C  C2  00BE   234          SUBL    #PTR_K_LENGTH,R8                        ; Start with this token
                     0F  11  00C1   235          BRB     10$                                     ; Process keywords
                                    00C3   236
                                    00C3   237 ;
                                    00C3   238 ; Process qualifier.
                                    00C3   239 ;
                     4C  10  00C3   240 7$:     BSBB    QUAL                                    ; Process qualifier
                   3F 50  E9  00C5   241          BLBC    R0,95$                                  ; Return false if not found
                53   01  D1  00C8   242          CMPL    #1,R3                                   ; Only one entity address?
                     3E  13  00CB   243          BEQL    97$                                     ; Yes, return
                     52  D6  00CD   244          INCL    R2                                      ; Incr the value level
                56   08  D0  00CF   245          MOVL    #8,R6                                   ; Set parameter count
                                    00D2   246
                                    00D2   247 ;
                                    00D2   248 ; Find the specified keyword entities.
                                    00D2   249 ;
        57   6542   DE  AA  C1  00D2   250 10$:     ADDL3   WRK_L_TAB_VEC(R10),(R5)[R2],R7          ; Get entity block address
                58   0C  C0  00D8   251 20$:     ADDL    #PTR_K_LENGTH,R8                        ; Get next PTR block
             68   04  1C  ED  00DB   252          CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; End of line?
                     04          00DF   253                  #PTR_K_ENDLINE                  ;
                     21  13  00E0   254          BEQL    85$                                     ; Yes, return false
                  56   06  A8  91  00E2   255          CMPB    PTR_B_PARMCNT(R8),R6                    ; Right parameter count?
                     1B  14  00E6   256          BGTR    85$                                     ; No, return false
                  52   04  A8  91  00E8   257          CMPB    PTR_B_LEVEL(R8),R2                      ; Value depth too shallow?
                     15  1F  00EC   258          BLSSU   85$                                     ; Yes, return false
                     E8  12  00EE   259          BNEQ    20$                                     ; Skip if not exact
             57   08  A8  D1  00F0   260          CMPL    PTR_L_ENTITY(R8),R7                     ; Is is the one we want?
                     E2  12  00F4   261          BNEQ    20$                                     ; No, get next PTR
          D8 52      53  F3  00F6   262          AOBLEQ  R3,R2,10$                               ; Yes, get next entity
          07 68      14  E0  00FA   263          BBS     #PTR_V_NEGATE,(R8),90$                  ; If negated, return false
                                    00FE   264
                                    00FE   265 ;
                                    00FE   266 ; Clean up and return
                                    00FE   267 ;
                50   01  D0  00FE   268 80$:     MOVL    #1,R0                                   ; Return true
                     04  11  0101   269          BRB     95$                                     ;
                                    0103   270
                     58  D4  0103   271 85$:     CLRL    R8                                      ; Mark entity not found
                     50  D4  0105   272 90$:     CLRL    R0                                      ; Return false
                52   8E  7D  0107   273 95$:     MOVQ    (SP)+,R2                                ; Restore R2,R3
                         05  010A   274          RSB                                             ; Return
                                    010B   275
          EF 68      14  E1  010B   276 97$:     BBC     #PTR_V_NEGATE,(R8),80$                  ; If not negated, return tru
                     F2  11  010F   277          BRB     85$                                     ; Else, return false
```

```
                     0111   279 ;++
                     0111   280 ; QUAL - Find qualifier on command line.
                     0111   281 ;
                     0111   282 ; Outputs:
                     0111   283 ;
                     0111   284 ;       R0 = true/false status
                     0111   285 ;       R8 = PTR block address
                     0111   286 ;--
                     0111   287
                     0111   288 QUAL:
                56   D4 0111   289         CLRL    R6                                      ; Assume qual not found
                     0113   290 ;        BBC     #ENT_V_PARM,ENT_W_FLAGS(R7),50$         ; Skip local search if unnec
                     0113   291
                     0113   292 ;
                     0113   293 ; Find local parameter context.
                     0113   294 ; (We are temporarily treating all qualifiers as global. Therefore, the following
                     0113   295 ; checks all of the local qualifiers on the command line for presence)
                     0113   296 ;
                     0113   297 ;       TSTL    R9                                      ; Are there any parameters?
                     0113   298 ;       BEQL    50$                                     ; No, then skip
                     0113   299 ;       MOVAB   WRK_G_RESULT-PTR_K_LENGTH(R10),R8       ; Get addr of PTR blocks
                     0113   300 ;       CLRL    R0                                      ; Init parameter count
                     0113   301 ;40$:   ADDL    #PTR_K_LENGTH,R8                        ; Get next PTR block
                     0113   302 ;       CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; Parameter value?
                     0113   303 ;               #PTR_K_PARAMETR                         ;
                     0113   304 ;       BNEQ    40$                                     ; No, get next PTR
                     0113   305 ;       INCL    R0                                      ; Incr param count
                     0113   306 ;       CMPL    R0,R9                                   ; Are we there yet?
                     0113   307 ;       BNEQ    40$                                     ; No, get next PTR
                     0113   308 ;
                     0113   309 ;
                     0113   310 ; Check for local qualifiers.
                     0113   311 ;
                     0113   312 ;45$:   ADDL    #PTR_K_LENGTH,R8                        ; Get next PTR block
                     0113   313 ;       CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; End of line or next
                     0113   314 ;               #PTR_K_PARAMETR                         ;   parameter value?
                     0113   315 ;       BGEQU   50$                                     ; Yes, then done
                     0113   316 ;       CMPL    PTR_L_ENTITY(R8),R7                     ; Is is the one we want?
                     0113   317 ;       BNEQ    45$                                     ; No, get next PTR
                     0113   318 ;       MOVL    R8,R6                                   ; Yes, save PTR
                     0113   319 ;       BRB     45$                                     ; Get next PTR
                     0113   320
                     0113   321
                     0113   322
                     0113   323 ;
                     0113   324 ; Check for global qualifiers.
                     0113   325 ;
          21 04 A7   08   E1 0113   326 50$:    BBC     #ENT_V_VERB,ENT_W_FLAGS(R7),40$         ; Skip global search if unne
                     0118   327 ;       TSTL    R6                                      ; Was qual found
                     0118   328 ;       BNEQ    40$                                     ; Yes, then skip
           58   F9AA CA   9E 0118   329         MOVAB   WRK_G_RESULT-PTR_K_LENGTH(R10),R8       ; Get addr of PTR blocks
                58   0C   C0 011D   330 55$:    ADDL    #PTR_K_LENGTH,R8                        ; Get next PTR block
           68   04   1C   ED 0120   331         CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; End of line?
                     04   0124   332                 #PTR_K_ENDLINE                          ;
                     12   13   0125   333         BEQL    40$                                     ; Yes, then done
           68   04   1C   ED 0127   334         CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; Command qualifier?
                     00        012B   335                 #PTR_K_COMDQUAL                         ;
```

```
                EF   12  012C    336          BNEQ    55$                                     ; No, get next PTR
          57 08 A8   D1  012E    337          CMPL    PTR_L_ENTITY(R8),R7                     ; Is is the one we want?
                E9   12  0132    338          BNEQ    55$                                     ; No, get next PTR
             56 58   D0  0134    339          MOVL    R8,R6                                   ; Yes, save PTR
                E4   11  0137    340          BRB     55$                                     ; Get next PTR
                         0139    341 ;
                         0139    342 ; Look at all the local qualifiers on the command line.
                         0139    343 ; (Checked after global qualifiers to establish a defined order)
                         0139    344 ;
       25 04 A7 09       E1  0139    345 40$:     BBC     #ENT_V_PARM,ENT_W_FLAGS(R7),60$         ; Skip local search if unnec
                59       D5  013E    346          TSTL    R9                                      ; Are there any parameters?
                21       13  0140    347          BEQL    60$                                     ; No, then skip
          58 F9AA CA       9E  0142    348          MOVAB   WRK_G_RESULT-PTR_K_LENGTH(R10),R8       ; Get addr of PTR blocks
                58       0C  C0  0147    349 45$:     ADDL    #PTR_K_LENGTH,R8                        ; Get next PTR block
          68 04 1C       ED  014A    350          CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; End of line?
                04             014E    351                  #PTR_K_ENDLINE                  ;
                12       13  014F    352          BEQL    60$                                     ; Yes, then done
          68 04 1C       ED  0151    353          CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; Parameter qualifier?
                01             0155    354                  #PTR_K_PARMQUAL                 ;
                EF   12  0156    355          BNEQ    45$                                     ; No, get next PTR
          57 08 A8   D1  0158    356          CMPL    PTR_L_ENTITY(R8),R7                     ; Is is the one we want?
                E9   12  015C    357          BNEQ    45$                                     ; No, get next PTR
             56 58   D0  015E    358          MOVL    R8,R6                                   ; Yes, save PTR
                E4   11  0161    359          BRB     45$                                     ; Get next PTR
                         0163    360
                         0163    361 ;
                         0163    362 ; Set up registers and return.
                         0163    363 ;
                50       D4  0163    364 60$:     CLRL    R0                                      ; Assume not found
          58    56       D0  0165    365          MOVL    R6,R8                                   ; Save found PTR address
                03       13  0168    366          BEQL    90$                                     ; Branch if none found
          50    01       D0  016A    367          MOVL    #1,R0                                   ; Set present
                         05  016D    368 90$:     RSB                                             ; Return
```

```
                          016E    370          .SBTTL  Evaluate Not Expression
                          016E    371 ;---
                          016E    372 ; NOT - Evaluate not expression
                          016E    373 ;
                          016E    374 ; This routine is called to evaluate a not expression.
                          016E    375 ;
                          016E    376 ; INPUTS:
                          016E    377 ;
                          016E    378 ;       R1 =    Address of expression block
                          016E    379 ;       R10 =   Address of command work area (WRK)
                          016E    380 ;
                          016E    381 ; OUTPUTS:
                          016E    382 ;
                          016E    383 ;       R0 =    Value of expression - true or false
                          016E    384 ;
                          016E    385 ;+++
                          016E    386
                          016E    387 NOT:
      DE AA   C1          016E    388          ADDL3   WRK_L_TAB_VEC(R10),-                    ; Get subexpression address
51    08 A1                 0171    389                  EXP_L_OPERAND_LIST(R1),R1               ;
       FEED   30            0174    390          BSBW    DISPATCH                               ; Evaluate the subexpression
         50   D6            0177    391          INCL    R0                                     ; Toggle the result
              05            0179    392          RSB                                            ; Return
                          017A    393
```

```
                     017A   395          .SBTTL  Evaluate Any2 Expression
                     017A   396 ;---
                     017A   397 ; ANY2 - Evaluate any2 expression
                     017A   398 ;
                     017A   399 ; This routine is called to evaluate a any2 expression.
                     017A   400 ;
                     017A   401 ; INPUTS:
                     017A   402 ;
                     017A   403 ;       R1 =    Address of expression block
                     017A   404 ;       R10 =   Address of command work area (WRK)
                     017A   405 ;
                     017A   406 ; OUTPUTS:
                     017A   407 ;
                     017A   408 ;       R0 =    Value of expression - true or false
                     017A   409 ;
                     017A   410 ;+++
                     017A   411
                     017A   412 ANY2:
      7E   52   7D   017A   413          MOVQ    R2,-(SP)                        ; Save R2,R3
           7E   D4   017D   414          CLRL    -(SP)                           ; Assume false
      53   51   D0   017F   415          MOVL    R1,R3                           ; Copy expression block addr
52    06 A3   3C   0182   416          MOVZWL  EXP_W_TRO_COUNT(R3),R2          ; Get operand count
     DE AA   C1   0186   417 10$:     ADDL3   WRK_L_TAB_VEC(R10),-            ; Get subexpression address
51   04 A342          0189   418                  EXP_L_OPERAND_LIST-4(R3)[R2],R1 ;
       FED4   30   018D   419          BSBW    DISPATCH                        ; Evaluate the subexpression
     04 50   E9   0190   420          BLBC    R0,20$                          ; Branch if false
05 6E    00   E2   0193   421          BBSS    #0,(SP),30$                     ; Branch if second true
     EC 52   F5   0197   422 20$:     SOBGTR  R2,10$                          ; Loop until done
        50   D4   019A   423          CLRL    R0                              ; Return false
        8E   D5   019C   424 30$:     TSTL    (SP)+                           ; Pop state
   52   8E   7D   019E   425          MOVQ    (SP)+,R2                        ; Restore R2,R3
              05   01A1   426          RSB                                     ; Return
                   01A2   427
```

```
                     01A2    429          .SBTTL  Evaluate And Expression
                     01A2    430 ;---
                     01A2    431 ; AND - Evaluate and expression
                     01A2    432 ;
                     01A2    433 ; This routine is called to evaluate a and expression.
                     01A2    434 ;
                     01A2    435 ; INPUTS:
                     01A2    436 ;
                     01A2    437 ;       R1 =    Address of expression block
                     01A2    438 ;       R10 =   Address of command work area (WRK)
                     01A2    439 ;
                     01A2    440 ; OUTPUTS:
                     01A2    441 ;
                     01A2    442 ;       R0 =    Value of expression - true or false
                     01A2    443 ;
                     01A2    444 ;+++
                     01A2    445
                     01A2    446 AND:
      7E   52   7D   01A2    447          MOVQ    R2,-(SP)                        ; Save R2,R3
      53   51   D0   01A5    448          MOVL    R1,R3                           ; Copy expression block addr
   52   06 A3   3C   01A8    449          MOVZWL  EXP_W_TRO_COUNT(R3),R2          ; Get operand count
      DE AA   C1   01AC    450 10$:     ADDL3   WRK_L_TAB_VEC(R10),-            ; Get subexpression address
51   04 A342        01AF    451                  EXP_L_OPERAND_LIST-4(R3)[R2],R1 ;
        FEAE   30   01B3    452          BSBW    DISPATCH                        ; Evaluate the subexpression
      06 50   E9   01B6    453          BLBC    R0,30$                          ; Branch if false
      F0 52   F5   01B9    454          SOBGTR  R2,10$                          ; Loop until done
      50   01   D0   01BC    455          MOVL    #1,R0                           ; Return true
      52   8E   7D   01BF    456 30$:     MOVQ    (SP)+,R2                        ; Restore R2,R3
                05   01C2    457          RSB                                     ; Return
                     01C3    458
```

```
                     01C3   460          .SBTTL  Evaluate Or Expression
                     01C3   461 ;---
                     01C3   462 ; OR - Evaluate or expression
                     01C3   463 ;
                     01C3   464 ; This routine is called to evaluate a or expression.
                     01C3   465 ;
                     01C3   466 ; INPUTS:
                     01C3   467 ;
                     01C3   468 ;       R1 =    Address of expression block
                     01C3   469 ;       R10 =   Address of command work area (WRK)
                     01C3   470 ;
                     01C3   471 ; OUTPUTS:
                     01C3   472 ;
                     01C3   473 ;       R0 =    Value of expression - true or false
                     01C3   474 ;
                     01C3   475 ;+++
                     01C3   476
                     01C3   477 OR:
      7E    52   7D  01C3   478          MOVQ    R2,-(SP)                        ; Save R2,R3
      53    51   D0  01C6   479          MOVL    R1,R3                           ; Copy expression block addr
   52    06 A1   3C  01C9   480          MOVZWL  EXP_W_TRO_COUNT(R1),R2          ; Get operand count
      DE AA   C1  01CD   481 10$:     ADDL3   WRK_L_TAB_VEC(R10),-            ; Get subexpression address
51    04 A342       01D0   482                  EXP_L_OPERAND_LIST-4(R3)[R2],R1 ;
        FE8D   30  01D4   483          BSBW    DISPATCH                        ; Evaluate the subexpression
      05 50   E8  01D7   484          BLBS    R0,30$                          ; Branch if false
      F0 52   F5  01DA   485          SOBGTR  R2,10$                          ; Loop until done
         50   D4  01DD   486          CLRL    R0                              ; Return false
      52   8E   7D  01DF   487 30$:     MOVQ    (SP)+,R2                        ; Restore R2,R3
                05  01E2   488          RSB                                     ; Return
                     01E3   489
```

```
                     01E3   491          .SBTTL  Evaluate Xor Expression
                     01E3   492 ;---
                     01E3   493 ; XOR - Evaluate xor expression
                     01E3   494 ;
                     01E3   495 ; This routine is called to evaluate a xor expression.
                     01E3   496 ;
                     01E3   497 ; INPUTS:
                     01E3   498 ;
                     01E3   499 ;       R1 =    Address of expression block
                     01E3   500 ;       R10 =   Address of command work area (WRK)
                     01E3   501 ;
                     01E3   502 ; OUTPUTS:
                     01E3   503 ;
                     01E3   504 ;       R0 =    Value of expression - true or false
                     01E3   505 ;
                     01E3   506 ;+++
                     01E3   507
                     01E3   508 XOR:
      7E   52   7D   01E3   509          MOVQ    R2,-(SP)                        ; Save R2,R3
           7E   D4   01E6   510          CLRL    -(SP)                           ; Assume false
      53   51   D0   01E8   511          MOVL    R1,R3                           ; Copy expression block addr
52    06 A3   3C   01EB   512          MOVZWL  EXP_W_TRO_COUNT(R3),R2          ; Get operand count
     DE AA   C1   01EF   513 10$:     ADDL3   WRK_L_TAB_VEC(R10),-            ; Get subexpression address
51   04 A342         01F2   514                  EXP_L_OPERAND_LIST-4(R3)[R2],R1 ;
        FE6B   30   01F6   515          BSBW    DISPATCH                        ; Evaluate the subexpression
      06 50   E9   01F9   516          BLBC    R0,20$                          ; Branch if false
         50   D4   01FC   517          CLRL    R0                              ; Set false
06 6E    00   E2   01FE   518          BBSS    #0,(SP),30$                     ; Branch if second true
     EA 52   F5   0202   519 20$:     SOBGTR  R2,10$                          ; Loop until done
      50   6E   D0   0205   520          MOVL    (SP),R0                         ; Return value
           8E   D5   0208   521 30$:     TSTL    (SP)+                           ; Pop state
52         8E   7D   020A   522          MOVQ    (SP)+,R2                        ; Restore R2,R3
                05   020D   523          RSB                                     ; Return
                     020E   524
```

```
                     020E    526          .SBTTL  Evaluate Neg Path Expression
                     020E    527 ;---
                     020E    528 ; NEG - Evaluate neg path expression
                     020E    529 ;
                     020E    530 ; This routine is called to evaluate a neg path expression.
                     020E    531 ;
                     020E    532 ; INPUTS:
                     020E    533 ;
                     020E    534 ;       R1 =    Address of expression block
                     020E    535 ;       R9 =    Number of parameter to conduct local qualifier search at
                     020E    536 ;       R10 =   Address of command work area (WRK)
                     020E    537 ;
                     020E    538 ; OUTPUTS:
                     020E    539 ;
                     020E    540 ;       R0 =    Value of expression - true or false
                     020E    541 ;       R5,R6 are trashed
                     020E    542 ;       R7 =    Address of ENT block associated with last
                     020E    543 ;               conflicting entity
                     020E    544 ;       R8 =    Address of PTR block associated with last
                     020E    545 ;               conflicting entity
                     020E    546 ;       R9 =    The number of the parameter providing the qualifier context
                     020E    547 ;
                     020E    548 ;+++
                     020E    549
                     020E    550 ;
                     020E    551 ; Init state for the search.
                     020E    552 ;
      DE AA   C1     020E    553 NEG:    ADDL3   WRK_L_TAB_VEC(R10),-                    ; Get subexpression address
   51   08 A1          0211    554                EXP_L_OPERAND_LIST(R1),R1               ;
     7E   52   7D      0214    555         MOVQ    R2,-(SP)                                ; Save R2,R3
     52   01   D0      0217    556         MOVL    #1,R2                                   ; Init value level
     53   06 A1   3C    021A    557         MOVZWL  EXP_W_TRO_COUNT(R1),R3                  ; Get operand count
   58   F9AA CA   9E    021E    558         MOVAB   WRK_G_RESULT-PTR_K_LENGTH(R10),R8       ; Get addr of PTR blocks
     55   04 A1   9E    0223    559         MOVAB   EXP_L_OPERAND_LIST-4(R1),R5             ; Get addr of entity block l
                         0227    560
                         0227    561 ;
                         0227    562 ; If the path specifies a parameter, find the first parameter value that
                         0227    563 ; could possibly be a value of the specified parameter.
                         0227    564 ;
57   6542   DE AA   C1     0227    565         ADDL3   WRK_L_TAB_VEC(R10),(R5)[R2],R7    ; Get entity block address
     03 A7   02   91       022D    566         CMPB    #ENT_K_QUALIFIER,ENT_B_SUBTYPE(R7)      ; Is it a qualifier?
             2A   13       0231    567         BEQL    7$                                      ; Yes, then process it
     56   14 A7   9A       0233    568         MOVZBL  ENT_B_NUMBER(R7),R6                     ; Get parameter number
                         0237    569
        58   0C   C0     0237    570 5$:     ADDL    #PTR_K_LENGTH,R8                        ; Get next PTR block
     68   04   1C   ED     023A    571         CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; End of line?
                04           023E    572                 #PTR_K_ENDLINE                  ;
                5D   13      023F    573         BEQL    85$                             ; Yes, return false
     68   04   1C   ED     0241    574         CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-          ; Parameter?
                03           0245    575                 #PTR_K_PARAMETR                 ;
                EF   12      0246    576         BNEQ    5$                              ; No, skip it
     56   06 A8   91       0248    577         CMPB    PTR_B_PARMCNT(R8),R6                    ; Is is the one we want?
                E9   12      024C    578         BNEQ    5$                              ; No, get next PTR
                         024E    579
                         024E    580 ;
                         024E    581 ; We are processing and have found a parameter specification.  If we are
                         024E    582 ; not looking for a specific keyword, return failure now.  If we are, then
```

```
                                024E    583 ; set up the index and ptr block starting address for the search.
                                024E    584 ;
                53    01    D1  024E    585         CMPL    #1,R3                   ; Only one entity address?
                      4B    13  0251    586         BEQL    85$                     ; Yes, return false
                      53    D7  0253    587         DECL    R3                      ; Put in synch with level
                55    04    C0  0255    588         ADDL    #4,R5                   ; Shift the table address
                58    0C    C2  0258    589         SUBL    #PTR_K_LENGTH,R8        ; Start with this token
                      10    11  025B    590         BRB     10$                     ; Process keywords
                                025D    591
                                025D    592 ;
                                025D    593 ; Process qualifier.
                                025D    594 ;
                  FEB1        30  025D    595 7$:     BSBW    QUAL                    ; Process qualifier
                3F    50      E9  0260    596         BLBC    R0,95$                  ; Return false if not found
                53    01      D1  0263    597         CMPL    #1,R3                   ; Only one entity address?
                      3E      13  0266    598         BEQL    97$                     ; Yes, return
                      52      D6  0268    599         INCL    R2                      ; Incr the value level
                56    08      D0  026A    600         MOVL    #8,R6                   ; Set parameter count
                                026D    601
                                026D    602 ;
                                026D    603 ; Find the specified keyword entities.
                                026D    604 ;
57    6542    DE    AA      C1  026D    605 10$:    ADDL3   WRK_L_TAB_VEC(R10),(R5)[R2],R7  ; Get entity block address
              58    0C      C0  0273    606 20$:    ADDL    #PTR_K_LENGTH,R8        ; Get next PTR block
            68    04      1C    ED  0276    607         CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R8),-  ; End of line?
                        04          027A    608                 #PTR_K_ENDLINE      ;
                        21    13    027B    609         BEQL    85$                 ; Yes, return false
                  56    06  A8    91  027D    610         CMPB    PTR_B_PARMCNT(R8),R6    ; Right parameter count?
                        1B    14    0281    611         BGTR    85$                 ; No, return false
                  52    04  A8    91  0283    612         CMPB    PTR_B_LEVEL(R8),R2  ; Value depth too shallow?
                        15    1F    0287    613         BLSSU   85$                 ; Yes, return false
                        E8    12    0289    614         BNEQ    20$                 ; Skip if not exact
              57      08  A8    D1  028B    615         CMPL    PTR_L_ENTITY(R8),R7     ; Is is the one we want?
                        E2    12    028F    616         BNEQ    20$                 ; No, get next PTR
            D8  52      53    F3  0291    617         AOBLEQ  R3,R2,10$               ; Yes, get next entity
            07  68      14    E1  0295    618         BBC     #PTR_V_NEGATE,(R8),90$  ; If not negated, return fal
                                0299    619
                                0299    620 ;
                                0299    621 ; Clean up and return
                                0299    622 ;
                50    01    D0  0299    623 80$:    MOVL    #1,R0                   ; Return true
                      04    11  029C    624         BRB     95$                     ;
                                029E    625
                      58    D4  029E    626 85$:    CLRL    R8                      ; Mark entity not found
                      50    D4  02A0    627 90$:    CLRL    R0                      ; Return false
                52    8E    7D  02A2    628 95$:    MOVQ    (SP)+,R2                ; Restore R2,R3
                            05  02A5    629         RSB                             ; Return
                                02A6    630
            EF  68      14    E0  02A6    631 97$:    BBS     #PTR_V_NEGATE,(R8),80$  ; If negated, return true
                      F2    11  02AA    632         BRB     85$                     ; Else, return false
                                02AC    633
                                02AC    634         .END
```

```
AND                             000001A2 R      02
ANY2                            0000017A R      02
CLI$_CONFLICT                   ******** X      02
CLI$_NORMAL                     ******** X      02
DCL$EVAL_DISALLOW               00000000 RG     02
DISPATCH                        00000064 R      02
ENT_B_NUMBER                  = 00000014
ENT_B_SUBTYPE                 = 00000003
ENT_K_QUALIFIER               = 00000002
ENT_V_PARM                    = 00000009
ENT_V_VERB                    = 00000008
ENT_W_FLAGS                   = 00000004
ENT_W_NAME                    = 00000016
EXP_B_SUBTYPE                 = 00000003
EXP_K_NEG                     = 00000007
EXP_K_PATH                    = 00000001
EXP_L_OPERAND_LIST            = 00000008
EXP_W_TRO_COUNT               = 00000006
NEG                             000002CE R      02
NOT                             0000016E R      02
OR                              000001C3 R      02
PATH                            0000007A R      02
PRC_B_CONTINUE                  000000F3
PRC_B_DEFRADIX                  000000AE
PRC_B_EXMDEPMOD                 000000AD
PRC_B_EXMDEPWID                 000000AC
PRC_B_EXONLYL                   0000012D
PRC_B_FLAGS2                    000000AF
PRC_B_IMGFLAG                   00000078
PRC_B_OUTFLAGS                  0000012C
PRC_B_PROMPTLEN                 000000F0
PRC_C_LENGTH                    00000534
PRC_G_COMMANDS                  00000133
PRC_G_PROMPT                    000000F4
PRC_K_LENGTH                    00000534
PRC_L_CURRKEY                   00000048
PRC_L_EXMDEPADR                 000000A8
PRC_L_EXTARG                    00000094
PRC_L_EXTBLK                    0000008C
PRC_L_EXTCOD                    0000009C
PRC_L_EXTHND                    00000090
PRC_L_EXTPRM                    00000098
PRC_L_IDFLNK                    000000BC
PRC_L_IMGACTSTS                 00000080
PRC_L_INDCLOCK                  0000007C
PRC_L_INDEPTH                   0000005C
PRC_L_INDFAB                    0000001C
PRC_L_INDINPRAB                 00000014
PRC_L_INDOUTRAB                 00000018
PRC_L_INPRAB                    00000008
PRC_L_LASTKEY                   0000004C
PRC_L_LSTSTATUS                 000000B0
PRC_L_ONCTLY                    000000B8
PRC_L_ONERROR                   0000006C
PRC_L_OUTOFBAND                 000000B4
PRC_L_OUTRAB                    0000000C
PRC_L_OUTRABCTX                 00000118
PRC_L_PPFLIST                   00000070
PRC_L_RECALLPTR                 0000012F
PRC_L_RESTART                   00000058
PRC_L_SAVAP                     00000000
PRC_L_SAVFP                     00000004
PRC_L_SEVERITY                  00000050
PRC_L_SPWN                      000000C0
PRC_L_STACKLM                   000000A4
PRC_L_STACKPT                   000000A0
PRC_L_STATUS                    00000054
PRC_L_STS                       00000084
PRC_L_STV                       00000088
PRC_L_SYMBOL                    00000060
PRC_L_TMBX                      00000074
PRC_L_TRMLIST                   00000010
PRC_Q_ALLOCREG                  00000020
PRC_Q_COMMAND                   000000E0
PRC_Q_FLUSHTIME                 000000D0
PRC_Q_GLOBAL                    00000028
PRC_Q_IMAGENAME                 000000D8
PRC_Q_KEYPAD                    00000040
PRC_Q_LABEL                     00000030
PRC_Q_LOCAL                     00000038
PRC_Q_SAVEPRIV                  000000E8
PRC_T_OUTDVI                    0000011C
PRC_W_ASTIOSB                   000000C6
PRC_W_ASTRETN                   000000C8
PRC_W_ASTSTATUS                 000000C4
PRC_W_ATTMBX                    0000007A
PRC_W_FLAGS                     00000068
PRC_W_INPCHAN                   00000064
PRC_W_ONLEVEL                   0000006A
PRC_W_OUTIFI                    00000114
PRC_W_OUTISI                    00000116
PRC_W_OUTMBXCHN                 000000CA
PRC_W_OUTMBXREF                 000000CE
PRC_W_OUTMBXSIZ                 000000CC
PRC_W_PMPTCTRL                  000000F1
PRC_W_WAITIOSB                  00000066
PTR_B_LEVEL                     00000004
PTR_B_NUMBER                    00000005
PTR_B_PARMCNT                   00000006
PTR_B_VALUE                     00000000
PTR_C_LENGTH                    0000000C
PTR_K_COMDQUAL                = 00000000
PTR_K_ENDLINE                 = 00000004
PTR_K_LENGTH                    0000000C
PTR_K_PARAMETR                = 00000003
PTR_K_PARMQUAL                = 00000001
PTR_L_DESCR                     00000000
PTR_L_ENTITY                    00000008
PTR_S_OFFSET                  = 0000000C
PTR_S_TYPE                    = 00000004
PTR_S_VALUE                   = 00000008
PTR_V_NEGATE                  = 00000014
PTR_V_OFFSET                  = 00000008
PTR_V_TYPE                    = 0000001C
```

```
PTR_V_VALUE                     = 00000000
QUAL                              00000111 R      02
WRK_B_CMDOPT                      FFFFFFC3
WRK_B_MAXPARM                     FFFFFFD0
WRK_B_MINPARM                     FFFFFFD1
WRK_B_PARMCNT                     FFFFFFCE
WRK_B_PARMSUM                     FFFFFFCF
WRK_B_RECALLCNT                   FFFFFFC5
WRK_B_VALLEV                      FFFFFFC4
WRK_B_VERBTYP                     FFFFFFC2
WRK_C_LENGTH                      FFFFF486
WRK_G_BUFFER                      FFFFF492
WRK_G_INPBUF                      FFFFF896
WRK_G_RESULT                      FFFFF9B6
WRK_K_LENGTH                      FFFFF486
WRK_L_CHARPTR                     FFFFF48E
WRK_L_DISALLOW                    FFFFFFE6
WRK_L_ERRORRTN                    FFFFF9AE
WRK_L_EXPANDPTR                   FFFFF486
WRK_L_IMAGE                       FFFFFFE2
WRK_L_MARKPTR                     FFFFF48A
WRK_L_PAROUT                      FFFFFFD2
WRK_L_PMPTADDR                    FFFFF9A2
WRK_L_PROMPTRTN                   FFFFF9A6
WRK_L_PROPTR                      FFFFFFC6
WRK_L_QUABLK                      FFFFFFCA
WRK_L_READRTN                     FFFFF9AA
WRK_L_RECALLPTR                   FFFFFFEA
WRK_L_RSLEND                      FFFFFFB6
WRK_L_RSLNXT                      FFFFFFBA
WRK_L_SAVAP                       FFFFFFF8
WRK_L_SAVFP                       FFFFFFFC
WRK_L_SAVSP                       FFFFFFF4
WRK_L_SIGNALRTN                   FFFFFFD6
WRK_L_SPECRTN                     FFFFF9B2
WRK_L_TAB_VEC                     FFFFFFDE
WRK_L_VERB                        FFFFFFBE
WRK_W_FLAGS                       FFFFFFF0
WRK_W_FLAGS2                      FFFFFFF2
WRK_W_IMGCHAN                     FFFFFFEE
WRK_W_PMPTLEN                     FFFFF99E
XOR                               000001E3 R      02
```

+----------------+
! 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	000002AC	(684.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	16	00:00:00.04	00:00:01.86
Command processing	122	00:00:00.72	00:00:08.80
Pass 1	194	00:00:06.26	00:00:21.95
Symbol table sort	0	00:00:00.69	00:00:01.57
Pass 2	111	00:00:01.69	00:00:04.81
Symbol table output	19	00:00:00.13	00:00:00.53
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	463	00:00:09.55	00:00:39.54

The working set limit was 1050 pages.
32656 bytes (64 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 456 non-local and 40 local symbols.
634 source lines were read in Pass 1, producing 14 object records in Pass 2.
29 pages of virtual memory were used to define 15 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
_255DUA28:[DCL.OBJ]DCL.MLB;1	6
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	9

558 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

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

INTDEF
SDL

COMMAND
LIS

DESCRVAL
LIS

CONVERT
LIS

CLIMAC
MAR

CLIMSG
LIS

CONNECT
LIS

DCLPARSE
LIS

CHARMANIP
LIS

EXAMDEP
LIS

EXIT
LIS

INTIMAGES
MAR

DCXSTART
LIS

CLIGBL
LIS

DISALLOW
LIS

CLINT
LIS

CANCEL
LIS