```
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
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
DDDDDDDDDDDD        CCCCCCCCCCCC  LLLLLLLLLLLLLLL
```

```
                                                        M  6
**FILE**ID**RPSUB









RRRRRRRR    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB
RRRRRRRR    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB
RR      RR  PP      PP  SS          UU      UU  BB      BB
RR      RR  PP      PP  SS          UU      UU  BB      BB
RR      RR  PP      PP  SS          UU      UU  BB      BB
RR      RR  PP      PP  SS          UU      UU  BB      BB
RRRRRRRR    PPPPPPPP      SSSSSS    UU      UU  BBBBBBBB
RRRRRRRR    PPPPPPPP      SSSSSS    UU      UU  BBBBBBBB
RR  RR      PP                  SS  UU      UU  BB      BB
RR  RR      PP                  SS  UU      UU  BB      BB
RR    RR    PP                  SS  UU      UU  BB      BB      ....
RR    RR    PP                  SS  UU      UU  BB      BB      ....
RR      RR  PP          SSSSSSSS    UUUUUUUUUU  BBBBBBBB        ....
RR      RR  PP          SSSSSSSS    UUUUUUUUUU  BBBBBBBB        ....

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            .TITLE  RPSUB - DCL RESULT PARSE SUBROUTINES
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;*****************************************************************************
0000    ;*                                                                           *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    ;*  ALL RIGHTS RESERVED.                                                     *
0000    ;*                                                                           *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    ;*  TRANSFERRED.                                                             *
0000    ;*                                                                           *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    ;*  CORPORATION                                                              *
0000    ;*                                                                           *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    ;*                                                                           *
0000    ;*                                                                           *
0000    ;*****************************************************************************
0000    ;
0000    ;
0000    ;++
0000    ; FACILITY:     STARLET DCL CLI
0000    ;
0000    ; ABSTRACT:     MISC SUBROUTINES
0000    ;
0000    ;
0000    ; ENVIRONMENT:  NATIVE MODE USER CODE
0000    ;
0000    ; AUTHOR:       W.H.BROWN, CREATION DATE:14-APR-77
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-003 PCG0003         Peter George            15-Feb-1983
0000    ;               Update to new structure level.
0000    ;               Handle larger PTR data structure.
0000    ;               Move DCL$CNVASCBIN to CONVERT.
0000    ;
0000    ;       V03-002 PCG0002         Peter George            14-Nov-1982
0000    ;               Call DCL$TRIM to process the numeric string
0000    ;               before converting it to an integer.
0000    ;
0000    ;       V03-001 PCG0001         Peter George            30-Sep-1982
0000    ;               Use new larger PTR data structure.
0000    ;--
```

```
    0000          .SBTTL  DECLARATIONS
    0000   ;
    0000   ; MACRO LIBRARY CALLS
    0000   ;
    0000
    0000          PRCDEF                          ; DEFINE PROCESS WORK AREA
    0000          WRKDEF                          ; DEFINE COMMAND WORK AREA
    0000          $$CLITABDEF                     ; DEFINE TABLE STRUCTURE
    0000          PTRDEF                          ; DEFINE RESULT PARSE DESCRIPTOR
    0000          RPWDEF                          ; RESULT PARSE WORK DEFINITIONS
    0000          PLMDEF                          ; PARAMETER LIMIT DEFINITIONS
    0000          $CLIDEF                         ; CLI DEFINITIONS
    0000          $CLIMSGDEF                      ; CLI MESSAGE DEFINITIONS
    0000
    0000
    0000   ;
    0000   ; OWN STORAGE:
    0000   ;
00000000          .PSECT  DCL$ZCODE       BYTE,RD,NOWRT
    0000
```

```
                    0000            
                    0000                    .DSABL  LSB
                    0000                    .SBTTL  GET QUALIFIER DESCRIPTOR BLOCK
                    0000       ;++
                    0000       ; FUNCTIONAL DESCRIPTION:
                    0000       ;
                    0000       ;       THIS ROUTINE IS CALLED TO LOCATE THE COMMAND QUALIFIER
                    0000       ;       DESCRIPTOR BLOCK FOR A SPECIFIC QUALIFIER.
                    0000       ;       ALTERNATE ENTRY TO CHECK THAT QUALIFIER IS A PARAMETER
                    0000       ;       QUALIFIER AS OPPOSED TO AN OUTPUT SPECIFIER.
                    0000       ;
                    0000       ; CALLING SEQUENCE:
                    0000       ;
                    0000       ;       BSB/JSB DCL$GETQUALDESC         ; GET QUALIFIER DESCRIPTOR
                    0000       ;       BSB/JSB DCL$GETPARMQUAL         ; GET PARAMETER QUALIFIER DESCRIPTOR
                    0000       ;
                    0000       ; INPUT PARAMETERS:
                    0000       ;
                    0000       ;       R1 IS THE CODE TO IDENTIFY THE QUALIFIER
                    0000       ;
                    0000       ; IMPLICIT INPUTS:
                    0000       ;
                    0000       ;       R8 = ADDRESS OF UTILITY BIT ARRAY
                    0000       ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                    0000       ;       R10 = ADDRESS OF WORK BLOCK
                    0000       ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                    0000       ;
                    0000       ; OUTPUT PARAMETERS:
                    0000       ;
                    0000       ;       R2 IS THE ADDRESS OF THE QUALIFIER DESCRIPTOR BLOCK
                    0000       ;
                    0000       ; COMPLETION CODES:
                    0000       ;
                    0000       ;       R0 = SUCCESS/FAIL DEPENDING OF WHETHER THE DESCRIPTOR WAS FOUND
                    0000       ;
                    0000       ; SIDE EFFECTS:
                    0000       ;
                    0000       ;       TOP LEVEL RETURN (RET) TAKEN IF SEARCH FAILS
                    0000       ;
                    0000       ;--
                    0000
                    0000                    .ENABL  LSB
                    0000
                    0000       DCL$GETPARMQUAL::                               ; GET A PARAMETER QUALIFIER DESCRIPTOR
                    0000       DCL$GETQUALDESC::                               ; FIND A QUALIFIER DESCRIPTOR
   50    51    D0   0000                    MOVL    R1,R0                   ; COPY QUALIFIER NUMBER
         19    13   0003                    BEQL    90$                     ; ZERO IS INVALID QUALIFIER NUMBER
52    CA AB    D0   0005                    MOVL    WRK_L_QUABLK(R11),R2    ; POINT AT START OF QUALIFIER BLOCKS
         13    13   0009                    BEQL    90$                     ; BR IF NONE
         0B    11   000B                    BRB     20$                     ; START OF SEARCH
      08 A2    D5   000D       10$.         TSTL    ENT_L_NEXT(R2)          ; TEST OFFSET TO NEXT
         0C    13   0010                    BEQL    90$                     ; BR IF THIS IS LAST
      08 A2    C1   0012                    ADDL3   ENT_L_NEXT(R2),-        ; FIND ADDRESS OF NEXT ENT BLOCK
52    DE AB         0015                            WRK_L_TAB_VEC(R11),R2   ;
      F2 50    F5   0018       20$:         SOBGTR  R0,10$                  ; COUNT DOWN QUALIFIER NUMBER
         50    D6   001B                    INCL    R0                      ; INDICATE DESCRIPTOR FOUND
               05   001D                    RSB                             ; BACK TO THE CALLER
```

```
     001E   132
     001E   133 90$:     SETSTAT INVQUALNUM               ; SET ERROR-INVALID QUALIFER NUMBER
  04 0023   134          RET                              ; GO BACK TO DISPATCHER
     0024   135
     0024   136          .DSABL  LSB
```

```
                     0024          .SBTTL  FIND COMMAND QUALIFIER
                     0024     ;++
                     0024     ; FUNCTIONAL DESCRIPTION:
                     0024     ;
                     0024     ;        THIS COROUTINE IS CALLED TO SEARCH FOR A
                     0024     ;        COMMAND QUALIFIER IN THE RANGE OF THE CURRENT COMMAND.
                     0024     ;        THE SEARCH IS DONE OUT TO THE FIRST PARAMETER APPEARING
                     0024     ;        IN THE COMMAND, THEN FROM THE START OF THE FIRST PARAMETER
                     0024     ;        IN THE RANGE OF THE CURRENT COMMAND TO THE END OF THE
                     0024     ;        RANGE OF THE CURRENT COMMAND.
                     0024     ;
                     0024     ; CALLING SEQUENCE:
                     0024     ;
                     0024     ;        BSB/JSB DCL$LOCCMDQUAL
                     0024     ;
                     0024     ; INPUT PARAMETERS:
                     0024     ;
                     0024     ;        R1 IS THE CODE OF THE QUALIFIER TO LOCATE
                     0024     ;
                     0024     ; IMPLICIT INPUTS:
                     0024     ;
                     0024     ;        R8 = ADDRESS OF UTILITY BIT ARRAY
                     0024     ;        R9 = ADDRESS OF REQUEST DESCRIPTOR
                     0024     ;        R10 = ADDRESS OF WORK BLOCK
                     0024     ;        R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     0024     ;
                     0024     ; OUTPUT PARAMETERS:
                     0024     ;
                     0024     ;        R4 IS RETURNED AS THE ADDRESS OF THE DESCRIPTOR IF FOUND
                     0024     ;        R5 IS THE INDEX TO THE DESCRIPTOR IF FOUND
                     0024     ;
                     0024     ; COMPLETION CODES:
                     0024     ;
                     0024     ;        R0 IS SET TRUE OR FALSE DEPENDING OF SUCCESS OF SEARCH
                     0024     ;
                     0024     ; SIDE EFFECTS:
                     0024     ;
                     0024     ;        REGISTERS R4, R5 & R6 ARE USED BY THIS ROUTINE
                     0024     ;        AND MUST BE PRESERVED ACCROSS COROUTINE RETURNS.
                     0024     ;
                     0024     ;--
                     0024
                     0024     DCL$FNDCMDQUAL::                                 ; LOCATE THE COMMAND QUALIFIER
     55    01    D0  0024             MOVL    #1,R5                    ; SET INDEX TO START SEARCH
56    40 AA    DE  0027             MOVAL   RPW_G_PRMLIM(R10),R6     ; START OF PARAMETER LIMIT DESCRIPTORS
           50    D4  002B     10$:    CLRL    R0                       ; ASSUME NO MORE QUALIFIERS
08 AA      55    91  002D             CMPB    R5,RPW_B_STRPARM(R10)    ; THIS DESCRIPTOR IN RANGE OF VERB
           0E    1F  0031             BLSSU   40$                      ; BR IF YES
           06    12  0033             BNEQ    30$                      ; BR IF WITHIN A PARAMETER
55    01 A6    9A  0035     20$:    MOVZBL  PLM_B_FSTDESC(R6),R5     ; SET INDEX OF PLACE TO START LOOKING
           1C    13  0039             BEQL    70$                      ; BR WHEN DONE
02 A6      55    91  003B     30$:    CMPB    R5,PLM_B_LSTDESC(R6)     ; IS THIS WITHIN THE CURRENT PARAMETER?
           11    1A  003F             BGTRU   60$                      ; BR IF OUT OF RANGE OF THIS PARAMETER
           3E    10  0041     40$:    BSBB    DCL$SETDESCADR           ; SET ADDRESS OF RESULT DESCRIPTOR
     04    1C    ED  0043             CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,- ; VIELD LIMITS FOR TYPE
     00    64        0046                     PTR_C_DESCR(R4),#PTR_K_COMDQUAL ; IF THIS A COMMAND QUALIFIER?
           04    12  0048             BNEQ    50$                      ; BR IF NO-CONTINUE SEARCH
```

```
          50   D6  004A            INCL    R0                      ; SET SUCCESS
          9E   16  004C            JSB     @(SP)+                  ; RETURN WITH QUALIFIER
          55   D6  004E     50$:   INCL    R5                      ; ADVANCE INDEX TO NEXT DESCRIPTOR
          D9   11  0050            BRB     10$                     ; CHECK AGAIN
     56   04   C0  0052     60$:   ADDL    #PLM_K_SIZE,R6          ; SET TO NEXT PARAMETER LIMIT DESCRIPTOR
          DE   11  0055            BRB     20$                     ; TRY NEXT PARAMETER
               05  0057     70$:   RSB                             ; RETURN WITH VALUE OR ZERO
```

```
                     0058          .SBTTL  EXTRACT RESULT DESCRIPTOR FIELDS
                     0058  ;++
                     0058  ; FUNCTIONAL DESCRIPTION:
                     0058  ;
                     0058  ;       THIS ROUTINE IS CALLED TO TAKE A RESULT DESCRIPTOR APART
                     0058  ;       AND RETURN ITS COMPONENT PART AS INDIVIDUAL VALUES.
                     0058  ;
                     0058  ; CALLING SEQUENCE:
                     0058  ;
                     0058  ;       BSB/JSB DCL$EXTNXTDESC          ; EXTRACT NEXT DESCRIPTOR
                     0058  ;       BSB/JSB DCL$GETEXTDESC          ; GET AND EXTRACT DESCRIPTOR
                     0058  ;       BSB/JSB DCL$EXTRSLDESC          ; EXTRACT RESULT DESCRIPTOR
                     0058  ;
                     0058  ; INPUT PARAMETERS:
                     0058  ;
                     0058  ;       AT EXTRSLDESC WITH R4 CONTAINS THE ADDRESS OF THE DESCRIPTOR
                     0058  ;       AT EXTNXTDESC WITH R6 CONTAINS THE ADDRESS OF THE PARAMETER
                     0058  ;           LIMIT DESCRIPTOR.
                     0058  ;
                     0058  ; IMPLICIT INPUTS:
                     0058  ;
                     0058  ;       R8 = ADDRESS OF UTILITY BIT ARRAY
                     0058  ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                     0058  ;       R10 = ADDRESS OF WORK BLOCK
                     0058  ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     0058  ;
                     0058  ; OUTPUT PARAMETERS:
                     0058  ;
                     0058  ;       R1 = TYPE
                     0058  ;       R2 = SIZE OR VALUE DEPENDING ON THE DESCRIPTOR
                     0058  ;       R3 = ADDRESS OF THE ITEM
                     0058  ;       R4 = ADDRESS OF DESCRIPTOR
                     0058  ;
                     0058  ; COMPLETION CODES:
                     0058  ;
                     0058  ;       R0 = SUCCESS/FAILURE DEPENDING ON RESULT OF SEARCH
                     0058  ;
                     0058  ;--
                     0058          .ENABL  LSB
                     0058
                     0058  DCL$EXTNXTDESC::                        ; EXTRACT NEXT COMPLETE DESCRIPTOR
                     0058          SETSTAT FAIL                    ; ASSUME WONT FIND ONE
     55    66    9A  005A          MOVZBL  PLM_B_NXTDESC(R6),R5    ; SET POINTER TO DESCRIPTOR
           21    13  005D          BEQL    40$                     ; BR IF PARAMETER SET IS MISSING
  02 A6    55    91  005F          CMPB    R5,PLM_B_LSTDESC(R6)    ; IS THIS IN RANGE OF CURRENT PARAMETER?
           1B    1A  0063          BGTRU   40$                     ; BR IF NO
           66    96  0065          INCB    PLM_B_NXTDESC(R6)       ; ADVANCE INDEX TO NEXT
           50    D6  0067          INCL    R0                      ; SET ANY SUCCESSFUL STATUS
                     0069  DCL$GETEXTDESC::                        ; GET AND EXTRACT NEXT DESCTRIPTOR
           16    10  0069          BSBB    DCL$SETDESCADR          ; SET ADDRESS OF RESULT DESCRIPTOR
                     006B  DCL$EXTRSLDESC::                        ; EXTRACT FOR RESULT DESCRIPTION
     0C    08    EF  006B          EXTZV   #PTR_V_OFFSET,#PTR_S_OFFSET,- ; START BIT AND SIZE OF OFFSET
     53    64        006E                  PTR_C_DESCR(R4),R3      ; GET OFFSET INTO R3
53 F492  CB43    9E  0070          MOVAB   WRK_G_BUFFER(R11)[R3],R3 ; FIND ADDRESS OF ITEM IN BUFFER
     08    00    EF  0076          EXTZV   #PTR_V_VALUE,#PTR_S_VALUE,- ; START BIT AND SIZE OF VALUE
     52    64        0079                  PTR_C_DESCR(R4),R2      ; GET VALUE INTO R2
     04    1C    EF  007B          EXTZV   #PTR_V_TYPE,#PTR_S_TYPE,- ; START BIT AND SIZE OF TYPE
```

```
                 51    64       007E    260                PTR_L_DESCR(R4),R1       ; GET TYPE INTO R1
                          05    0080    261 40$:    RSB                             ;
                                0081    262
                                0081    263         .DSABL  LSB
```

```
                                  0081           .SBTTL  SET RESULT DESCRIPTOR ADDRESS
                                  0081  ;++
                                  0081  ; FUNCTIONAL DESCRIPTION:
                                  0081  ;
                                  0081  ;        THIS ROUTINE IS CALLED TO SET THE ADDRESS OF A RESULT
                                  0081  ;        DESCRIPTOR INTO R4.
                                  0081  ;
                                  0081  ; CALLING SEQUENCE:
                                  0081  ;
                                  0081  ;        BSB/JSB DCL$SETDESCADR
                                  0081  ;
                                  0081  ; INPUT PARAMETERS:
                                  0081  ;
                                  0081  ;        R5 CONTAINS THE INDEX FOR THE DESIRED DESCRIPTOR
                                  0081  ;
                                  0081  ; IMPLICIT INPUTS:
                                  0081  ;
                                  0081  ;        R10 = ADDRESS OF WORK BLOCK
                                  0081  ;        R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                                  0081  ;
                                  0081  ; OUTPUT PARAMETERS:
                                  0081  ;
                                  0081  ;        R4 IS LOADED WITH THE ADDRESS OF THE DESCRIPTOR
                                  0081  ;
                                  0081  ;--
                                  0081
                                  0081  DCL$SETDESCADR::
          54    55    0C    C5    0081           MULL3   #PTR_C_LENGTH,R5,R4                     ; GET BYTE OFFSET OF DESCRIP
      54    F9AA  CB44      9E    0085           MOVAB   WRK_G_RESULT-PTR_C_LENGTH(R11)[R4],R4   ; GET ADDRESS OF DESCRIPTOR
                            05    008B           RSB
```

```
                    008C          .SBTTL  GET PARAMETER
                    008C     ;++
                    008C     ; FUNCTIONAL DESCRIPTION:
                    008C     ;
                    008C     ;        THIS ROUTINE IS CALLED TO SEARCH THE RESULT DESCRIPTOR
                    008C     ;        BUFFER FOR THE NEXT OCCURANCE OF A PRARMETER
                    008C     ;
                    008C     ; CALLING SEQUENCE:
                    008C     ;
                    008C     ;        BSB/JSB DCL$GETPARM              ; GET A PARAMETER
                    008C     ;
                    008C     ; INPUT PARAMETERS:
                    008C     ;
                    008C     ;        R5 CONTAINS THE INDEX OF NEXT DESCRIPTOR TO CHECK
                    008C     ;
                    008C     ; IMPLICIT INPUTS:
                    008C     ;
                    008C     ;        R8 = ADDRESS OF UTILITY BIT ARRAY
                    008C     ;        R9 = ADDRESS OF REQUEST DESCRIPTOR
                    008C     ;        R10 = ADDRESS OF WORK BLOCK
                    008C     ;        R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                    008C     ;
                    008C     ; OUTPUT PARAMETERS:
                    008C     ;
                    008C     ;        R1 CONTAINS THE TYPE OF THE DESCRIPTOR(IE:PTR_K_PARAMETR)
                    008C     ;        R2 CONTAINS THE SIZE OF THE PARAMETER
                    008C     ;        R3 CONTAINS THE PRECEEDING TERMINATOR
                    008C     ;        R4 CONTAINS THE ADDRESS OF THE PARAMETER DESCRIPTOR
                    008C     ;        R5 IS THE INDEX TO THE DESCRIPTOR
                    008C     ;
                    008C     ; COMPLETION CODES:
                    008C     ;
                    008C     ;        R0 = SUCCESS/FAIL DEPENDING ON THE RESULT OF THE SEARCH
                    008C     ;
                    008C     ;--
                    008C
                    008C     DCL$GETPARM::                            ; GET THE NEXT PARAMETER
                    008C             SETSTAT FAIL                     ; ASSUME NO MORE PARAMETERS
         55    D6   008E     10$:    INCL    R5                       ; ADVANCE INDEX
         D7    10   0090             BSBB    DCL$GETEXTDESC           ; GET DESCRIPTOR AND EXTRACT FIELDS
     04  18    EF   0092             EXTZV   #PTR_V_TERM,#PTR_S_TERM,- ; GET THE TERMINATOR FORM THE PRVIOUS
 53   F4 A4         0095                     -PTR_C_LENGTH(R4),R3     ; DESCRIPTOR AND SAVE IN R3
     04  51    91   0098             CMPB    R1,#PTR_K_ENDLINE        ; IS THIS THE END OF LINE?
         08    13   009B             BEQL    30$                      ; NO MORE PARAMETERS
     03  51    91   009D             CMPB    R1,#PTR_K_PARAMETR       ; IS THE CURRENT A PARAMETER?
         EC    12   00A0             BNEQ    10$                      ; BR IF NO-TRY NEXT
                    00A2             SETSTAT SUCCESS                  ; SET FOUND ONE
               05   00A5     30$:    RSB                              ; RETURN TO CALLER
```

```
                           00A6             .SBTTL  RESULT PARSE INIT
                           00A6       ;++
                           00A6       ; FUNCTIONAL DESCRIPTION:
                           00A6       ;
                           00A6       ;         THIS ROUTINE IS CALLED TO ESTABLISH INITIAL CONDITIONS
                           00A6       ;         IN THE RESULT PARSE WORK AREA PRIOR TO PERFORMAING
                           00A6       ;         A RESULT PARSE.
                           00A6       ;
                           00A6       ; CALLING SEQUENCE:
                           00A6       ;
                           00A6       ;         ENTERED VIA A CASE FOLLOWED BY A CALL
                           00A6       ;
                           00A6       ; IMPLICIT INPUTS:
                           00A6       ;
                           00A6       ;         R9 = ADDRESS OF REQUEST DESCRIPTOR
                           00A6       ;         R10 = ADDRESS OF WORK BLOCK
                           00A6       ;         R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                           00A6       ;
                           00A6       ; OUTPUT PARAMETERS:
                           00A6       ;
                           00A6       ;         THE RESULT PARSE WORK AREA IS INITED
                           00A6       ;
                           00A6       ; COMPLETION CODES:
                           00A6       ;
                           00A6       ;         R0 = SUCCESS
                           00A6       ;
                           00A6       ;--
                           00A6
                           00A6      DCL$RPINIT::                                       ; RESULT APRSE INIT
                 62   10   00A6             BSBB    DCL$GETDCLWRK                       ; GET POINTER TO DCL PHASE 1 WORK AREA
   00    6A    00   2C     00A8             MOVC5   #0,(R10),#0,-                       ; SOURCE BUFFER SIZE AND FILL OF 0 TO
04 AA    007C  8F          00AC                     #CLI$C_WORKAREA-4,4(R10)           ; ZERO OUT THE WORK AREA
      04 AA    5B   D0     00B1             MOVL    R11,RPW_L_DCLWRK(R10)               ; SAVE WORK AREA ADDRESS FOR LATER
      57    C6 AB   D0     00B5             MOVL    WRK_L_PROPTR(R11),R7                ; GET ADDRESS OF COMMAND PROMPT TABLE
      56    40 AA   DE     00B9             MOVAL   RPW_G_PRMLIM(R10),R6                ; GET ADDRESS OF FIRST PARAM LIMIT TABLE
               55   D4     00BD             CLRL    R5                                  ; INIT INDEX TO MINUS FIRST DESCRIPTOR
               CB   10     00BF             BSBB    DCL$GETPARM                         ; RETREIVE THE NEXT(FIRST)PARAMETER
      08 AA    55   90     00C1             MOVB    R5,RPW_B_STRPARM(R10)               ; SAVE THE INDEX TO THE FIRST PARAMETER
         3A    50   E9     00C5             BLBC    R0,90$                              ; BR IF NO PARAMETERS IN COMMAND
      01 A6    55   90     00C8      10$:   MOVB    R5,PLM_B_FSTDESC(R6)                ; SET FIRST PARAMETER IN THIS LIST
         66    55   90     00CC             MOVB    R5,PLM_B_NXTDESC(R6)                ; ALSO THE NEXT TO PROCESS
               BB   10     00CF      20$:   BSBB    DCL$GETPARM                         ; LOCATE THE NEXT PARAMETER IN THE COMD
02 A6    55    01   83     00D1             SUBB3   #1,R5,PLM_B_LSTDESC(R6)             ; SET INDEX OF THE LAST PARAMETER SEEN
         29    50   E9     00D6             BLBC    R0,90$                              ; BR IF NO MORE
      01       53   91     00D9             CMPB    R3,#PTR_K_BLANK                     ; IS THIS THE START OF A PARMETER LIST?
               16   13     00DC             BEQL    50$                                 ; BR IF YES
               07   E0     00DE             BBS     #ENT_V_IMPCAT,-                     ; IF COMMAND HAS IMPLIED CONCATONATION
      EC 04    A7          00E0                     ENT_W_FLAGS(R7),20$                 ; THEN KEEP LOOKING TILL END OF PARAMETER
      05       53   91     00E3             CMPB    R3,#PTR_K_COMMA                     ; ELSE LOOK FOR PARAMETER LIST SEPARATOR
               E7   12     00E6             BNEQ    20$                                 ; IF NO A SEPARATOR, CONTINUE SCAN
      01       53   91     00E8      40$:   CMPB    R3,#PTR_K_BLANK                     ; SCAN FOR LAST PARAMETER IN THE LIST
               07   13     00EB             BEQL    50$                                 ; BR IF FOUND LAST PARAMETER
               9D   10     00ED             BSBB    DCL$GETPARM                         ; SEARCH FOR NEXT PARAMETER
         10    50   E9     00EF             BLBC    R0,90$                              ; BR IF RAN OUT OF PARAMETERS
               F4   11     00F2             BRB     40$                                 ; CHECK FOR BLANKS
         08    A7   C1     00F4      50$:   ADDL3   ENT_L_NEXT(R7),-                    ; SKIP TO NEXT DESCRIPTOR
   57       DE AB          00F7                     WRK_L_TAB_VEC(R11),R7               ;
```

```
       03 A6    55   90  00FA           MOVB    R5,PLM_B_TRMDESC(R6)    ; SAVE DESCRIPTOR OF PARAMETER TERMINATOR
                86   D5  00FE           TSTL    (R6)+                   ; POINT AT NEXT PARAMETER LIMIT DESCRIPTOR
                C6   11  0100           BRB     10$                     ; SCAN NEXT PARAMETER
                         0102      90$: SETSTAT SUCCESS                 ; SET ALL IS GOOD
       03 A6    55   90  0105           MOVB    R5,PLM_B_TRMDESC(R6)    ; SAVE FINAL TERMINATOR
                     04  0109           RET                             ; RETURN TO DISPATCHER
                         010A      ;
                         010A      ; SET WORK ADDRESS
                         010A      ;
                         010A      DCL$GETDCLWRK::                      ;
     00000000'EF     16  010A           JSB     CLI$GET_PRC             ; GET ADDRESS OF CLI PROCESS WORK AREA
       5B    04 AB   D0  0110           MOVL    PRC_L_SAVFP(R11),R11    ; GET ADDRESS OF COMMAND WORK AREA
                     05  0114           RSB                             ; RETURN TO CALLER
                         0115           .END
```

```
CLI$C_WORKAREA                 = 00000080                    PRC_L_OUTRABCTX                     00000118
CLI$GET_PRC                      ********    X     02        PRC_L_PPFLIST                       00000070
CLI$_INVQUALNUM                = 0003881A                    PRC_L_RECALLPTR                     0000012F
DCL$EXTNXTDESC                   00000058 RG       02        PRC_L_RESTART                       00000058
DCL$EXTRSLDESC                   0000006B RG       02        PRC_L_SAVAP                         00000000
DCL$FNDCMDQUAL                   00000024 RG       02        PRC_L_SAVFP                         00000004
DCL$GETDCLWRK                    0000010A RG       02        PRC_L_SEVERITY                      00000050
DCL$GETEXTDESC                   00000069 RG       02        PRC_L_SPWN                          000000C0
DCL$GETPARM                      0000008C RG       02        PRC_L_STACKLM                       000000A4
DCL$GETPARMQUAL                  00000000 RG       02        PRC_L_STACKPT                       000000A0
DCL$GETQUALDESC                  00000000 RG       02        PRC_L_STATUS                        00000054
DCL$RPINIT                       000000A6 RG       02        PRC_L_STS                           00000084
DCL$SETDESCADR                   00000081 RG       02        PRC_L_STV                           00000088
ENT_L_NEXT                     = 00000008                    PRC_L_SYMBOL                        00000060
ENT_V_IMPCAT                   = 00000007                    PRC_L_TMBX                          00000074
ENT_W_FLAGS                    = 00000004                    PRC_L_TRMLIST                       00000010
PLM_B_FSTDESC                    00000001                    PRC_Q_ALLOCREG                      00000020
PLM_B_LSTDESC                    00000002                    PRC_Q_COMMAND                       000000E0
PLM_B_NXTDESC                    00000000                    PRC_Q_FLUSHTIME                     000000D0
PLM_B_QUADESC                    00000003                    PRC_Q_GLOBAL                        00000028
PLM_B_TRMDESC                    00000003                    PRC_Q_IMAGENAME                     000000D8
PLM_C_SIZE                       00000004                    PRC_Q_KEYPAD                        00000040
PLM_K_SIZE                       00000004                    PRC_Q_LABEL                         00000030
PRC_B_CONTINUE                   000000F3                    PRC_Q_LOCAL                         00000038
PRC_B_DEFRADIX                   000000AE                    PRC_Q_SAVEPRIV                      000000E8
PRC_B_EXMDEPMOD                  000000AD                    PRC_T_OUTDVI                        0000011C
PRC_B_EXMDEPWID                  000000AC                    PRC_W_ASTIOSB                       000000C6
PRC_B_EXONLYL                    0000012D                    PRC_W_ASTRETN                       000000C8
PRC_B_FLAGS2                     000000AF                    PRC_W_ASTSTATUS                     000000C4
PRC_B_IMGFLAG                    00000078                    PRC_W_ATTMBX                        0000007A
PRC_B_OUTFLAGS                   0000012C                    PRC_W_FLAGS                         00000068
PRC_B_PROMPTLEN                  000000F0                    PRC_W_INPCHAN                       00000064
PRC_C_LENGTH                     00000534                    PRC_W_ONLEVEL                       0000006A
PRC_G_COMMANDS                   00000133                    PRC_W_OUTIFI                        00000114
PRC_G_PROMPT                     000000F4                    PRC_W_OUTISI                        00000116
PRC_K_LENGTH                     00000534                    PRC_W_OUTMBXCHN                     000000CA
PRC_L_CURRKEY                    00000048                    PRC_W_OUTMBXREF                     000000CE
PRC_L_EXMDEPADR                  000000A8                    PRC_W_OUTMBXSIZ                     000000CC
PRC_L_EXTARG                     00000094                    PRC_W_PMPTCTRL                      000000F1
PRC_L_EXTBLK                     0000008C                    PRC_W_WAITIOSB                      00000066
PRC_L_EXTCOD                     0000009C                    PTR_B_LEVEL                         00000004
PRC_L_EXTHND                     00000090                    PTR_B_NUMBER                        00000005
PRC_L_EXTPRM                     00000098                    PTR_B_PARMCNT                       00000006
PRC_L_IDFLNK                     000000BC                    PTR_B_VALUE                         00000000
PRC_L_IMGACTSTS                  00000080                    PTR_C_LENGTH                        0000000C
PRC_L_INDCLOCK                   0000007C                    PTR_K_BLANK                       = 00000001
PRC_L_INDEPTH                    0000005C                    PTR_K_COMDQUAL                    = 00000000
PRC_L_INDFAB                     0000001C                    PTR_K_COMMA                       = 00000005
PRC_L_INDINPRAB                  00000014                    PTR_K_ENDLINE                     = 00000004
PRC_L_INDOUTRAB                  00000018                    PTR_K_LENGTH                        0000000C
PRC_L_INPRAB                     00000008                    PTR_K_PARAMETR                    = 00000003
PRC_L_LASTKEY                    0000004C                    PTR_L_DESCR                         00000000
PRC_L_LSTSTATUS                  000000B0                    PTR_L_ENTITY                        00000008
PRC_L_ONCTLY                     000000B8                    PTR_S_OFFSET                      = 0000000C
PRC_L_ONERROR                    0000006C                    PTR_S_TERM                        = 00000004
PRC_L_OUTOFBAND                  000000B4                    PTR_S_TYPE                        = 00000004
PRC_L_OUTRAB                     0000000C                    PTR_S_VALUE                       = 00000008
```

```
PTR_V_OFFSET                    = 00000008
PTR_V_TERM                      = 00000018
PTR_V_TYPE                      = 0000001C
PTR_V_VALUE                     = 00000000
RPW_B_LSTDESC                     00000009
RPW_B_STRPARM                     00000008
RPW_C_HDRSIZ                      00000040
RPW_C_LENGTH                      00000080
RPW_G_BITS                        00000020
RPW_G_PRMLIM                      00000040
RPW_K_HDRSIZ                      00000040
RPW_K_LENGTH                      00000080
RPW_L_DCLWRK                      00000004
RPW_L_USERCTX                     00000000
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
```

```
                                        +----------------+
                                        ! 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                       00000115  (  277.)  02 (  2.)  NOPIC    USR    CON    REL     LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

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

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                    9      00:00:00.11    00:00:00.42
Command processing               83      00:00:00.81    00:00:05.05
Pass 1                          250      00:00:09.13    00:00:22.27
Symbol table sort                 0      00:00:01.20    00:00:02.81
Pass 2                           72      00:00:01.69    00:00:06.40
Symbol table output              20      00:00:00.16    00:00:00.68
Psect synopsis output             2      00:00:00.02    00:00:00.02
Cross-reference output            0      00:00:00.00    00:00:00.00
Assembler run totals            436      00:00:13.14    00:00:37.67

The working set limit was 1200 pages.
45492 bytes (89 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 815 non-local and 18 local symbols.
415 source lines were read in Pass 1, producing 13 object records in Pass 2.
34 pages of virtual memory were used to define 20 macros.

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

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[SYSLIB]SYSBLDMLB.MLB;1                 0
_$255$DUA28:[DCL.OBJ]DCL.MLB;1                      8
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                      0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                   4
TOTALS (all libraries)                             12

956 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

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

0073 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY