```
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
```

```
RRRRRRRR    PPPPPPPP    DDDDDDDD      CCCCCCCC  LL
RRRRRRRR    PPPPPPPP    DDDDDDDD      CCCCCCCC  LL
RR      RR  PP      PP  DD      DD  CC          LL
RR      RR  PP      PP  DD      DD  CC          LL
RR      RR  PP      PP  DD      DD  CC          LL
RR      RR  PP      PP  DD      DD  CC          LL
RRRRRRRR    PPPPPPPP    DD      DD  CC          LL
RRRRRRRR    PPPPPPPP    DD      DD  CC          LL
RR  RR      PP          DD      DD  CC          LL
RR  RR      PP          DD      DD  CC          LL
RR    RR    PP          DD      DD  CC          LL              ....
RR    RR    PP          DD      DD  CC          LL              ....
RR      RR  PP          DDDDDDDD      CCCCCCCC  LLLLLLLLLL      ....
RR      RR  PP          DDDDDDDD      CCCCCCCC  LLLLLLLLLL      ....

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

Table of contents

```
0000     1          .TITLE  RPDCL   - RESULT PARSE MAIN ROUTINE
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
0000    29  ;++
0000    30  ; FACILITY:     STARLET DCL CLI
0000    31  ;
0000    32  ; ABSTRACT:     RESULT PARSE MAIN ROUTINE
0000    33  ;
0000    34  ; ENVIRONMENT:  NATIVE MODE USER CODE
0000    35  ;
0000    36  ; AUTHOR:       W.H.BROWN, CREATION DATE:  13-APR-77
0000    37  ;
0000    38  ; MODIFIED BY:
0000    39  ;
0000    40  ;       V03-004 PCG0005         Peter George            30-Apr-1983
0000    41  ;               Change BSBW to JSB.
0000    42  ;
0000    43  ;       V03-003 PCG0004         Peter George            15-Feb-1983
0000    44  ;               Convert to new stucture level.
0000    45  ;               Reference qualifier number by PTR_B_NUMBER.
0000    46  ;
0000    47  ;       V03-002 PCG0003         Peter George            15-Nov-1982
0000    48  ;               Use DCL$CNVNOEDIT instead of DCL$CNVNUMDEC.
0000    49  ;               Recognize NEXTQUAL callbacks.
0000    50  ;
0000    51  ;       V03-001 PCG0002         Peter George            30-Sep-1982
0000    52  ;               Modify DCL$GETOPT to correctly check for syntax
0000    53  ;               changing entity.  Refer to PTR length symbolically.
0000    54  ;--
```

```
                     0000     56          .SBTTL  DECLARATIONS
                     0000     57 ;
                     0000     58 ; MACRO LIBRARY CALLS
                     0000     59 ;
                     0000     60          PRCDEF                          ; DEFINE PROCESS WORK AREA
                     0000     61          WRKDEF                          ; DEFINE COMMAND WORK AREA
                     0000     62          $$CLITABDEF                     ; DEFINE TABLE STRUCTURES
                     0000     63          PTRDEF                          ; DEFINE RESULT PARSE DESCRIPTOR FORMAT
                     0000     64          RPWDEF                          ; RESULT PARSE WORK DEFINITIONS
                     0000     65          PLMDEF                          ; PARAMETER LIMIT DEFINITIONS
                     0000     66          $CLIDEF                         ; CLI DEFINITIONS
                     0000     67          $CLIVERBDEF                     ; VERB TYPE CODES
                     0000     68          $CLIMSGDEF                      ; CLI MESSAGE DEFINITIONS
                     0000     69
                     0000     70 ;
                     0000     71 ; UTILITY CALL PARAMETER OFFSETS
                     0000     72 ;
00000004             0000     73          RQDESC  =       4               ; OFFSET TO REQUEST DESCRIPTOR
00000008             0000     74          RQWORK  =       8               ; OFFSET TO WORK BLOCK
0000000C             0000     75          RQBITS  =       12              ; OFFSET TO BIT ARRAY ADDRESS
                     0000     76 ;
                     0000     77 ; INTERNAL ERROR BIT - DON'T USE R5 AS RESULT DESCRIPTOR INDEX
                     0000     78 ;
0000001F             0000     79          INTERROR = 31                   ; BIT 31 FLAGS INTERNAL ERROR
                     0000     80
                00000000     81          .PSECT  DCL$ZCODE       BYTE,RD,NOWRT
```

```
                        0000     83          .SBTTL  RESULT PARSE INITIAL ENTRY
                        0000     84 ;++
                        0000     85 ; FUNCTIONAL DESCRIPTION:
                        0000     86 ;
                        0000     87 ;       THIS IS THE ENTRY POINT USED FOR ALL UTILITY SEVICE
                        0000     88 ;       CALL BACK REQUEST FOR SERVICE.
                        0000     89 ;
                        0000     90 ; CALLING SEQUENCE:
                        0000     91 ;
                        0000     92 ;       CALL    DCL$UTILSERV
                        0000     93 ;
                        0000     94 ; INPUT PARAMETERS:
                        0000     95 ;
                        0000     96 ;       RQDESC(AP) IS THE ADDRESS OF THE REQUEST DESCRIPTOR
                        0000     97 ;       RQWORK(AP) IS THE ADDRESS OF A WORK AREA FOR RESULT PARSE DATA
                        0000     98 ;       RQBITS(AP) IS THE ADDRESS OF A BIT ARRAY FOR INPUT/OUTPUT
                        0000     99 ;               PARAMETER DEFINITION REQUESTS
                        0000    100 ;
                        0000    101 ; OUTPUT PARAMETERS:
                        0000    102 ;
                        0000    103 ;       THE FUNCTION IS PURFORMED, OR AN ERROR IS RETURNED
                        0000    104 ;
                        0000    105 ; COMPLETION CODES:
                        0000    106 ;
                        0000    107 ;       R0 = SUCCESS/FAILURE DEPENDING ON RESULT OF SEARCH
                        0000    108 ;
                        0000    109 ;--
                OFFC    0000    110          .ENTRY  DCL$UTLSERV,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                        0002    111
    59   04 AC    D0    0002    112          MOVL    RQDESC(AP),R9           ; ADDRESS OF THE REQUEST DESCRIPTOR
       04   04    ED    0006    113          CMPZV   #CLI$V_PRITYP,#CLI$S_PRITYP,-   ; THIS IS A REQUEST FOR A CLINT SERV
       05   69          0009    114                  CLI$B_RQTYPE(R9),#CLI$K_CLINT
            03    12    000B    115          BNEQ    3$                      ; BRANCH IF NO
          00AA    31    000D    116          BRW     CLINT
       69   05    91    0010    117 3$:      CMPB    #CLI$K_CLISERV,CLI$B_RQTYPE(R9) ; THIS IS A REQUEST FOR A CLI SERVIC
            05    12    0013    118          BNEQ    5$                      ; BR IF NO
         01 A9    BE    0015    119          CHMS    CLI$W_SERVCOD(R9)       ; THIS MUST BE DONE IN SUPER MODE
            6F    11    0018    120          BRB     RET0                    ; RETURN TO REQUESTOR
    5A   08 AC    D0    001A    121 5$:      MOVL    RQWORK(AP),R10          ; GET ADDRESS OF WORK AREA
                        001E    122          IFNORD  #4,RPW_L_DCLWRK(R10),10$ ; CHECK IF WORK AREA IS ACCESSABLE
    5B   04 AA    D0    0025    123          MOVL    RPW_L_DCLWRK(R10),R11   ; IF YES-GET THE WORK AREA
         03 A9    94    0029    124 10$:     CLRB    CLI$B_RQSTAT(R9)        ; INITIAL RETURN STATUS FLAGS
         08 A9    7C    002C    125          CLRQ    CLI$Q_RQDESC(R9)        ; AND ZERO THE PARAMETER DESCRIPTOR
                        002F    126 ;
                        002F    127 ; FROM THIS POINT ON, R5 MUST ALWAYS CONTAIN THE TOKEN DESCRIPTOR
                        002F    128 ; OF THE CURRENT TOKEN BEING PARSED SO THAT ERROR REPORTING WORKS.
                        002F    129 ;
       55   01    D0    002F    130          MOVL    #1,R5                   ; SET DEFAULT RESULT PARSE INDEX
  8A'AF     6C    FA    0032    131          CALLG   (AP),B^RSLTPRS          ; CALL FOR ERROR FRAME
50  00030000 8F    C8    0036    132          BISL    #<CLI$_ABKEYW&^XOFFF0000>,R0 ; INCLUDE SUBSYSTEM NUMBER
         49 50    E8    003D    133          BLBS    R0,RET0                 ; BR IF NO ERRORS
                        0040    134 ;
                        0040    135 ; CALL ERROR ACTION ROUTINE WITH ERROR.  IF THE TOP BIT OF R0 IS SET,
                        0040    136 ; THEN R2/R3 CONTAIN THE DESCRIPTOR OF THE TOKEN IN ERROR.  IF R5 IS
                        0040    137 ; NON-ZERO, IT CONTAINS THE TOKEN INDEX OF THE TOKEN IN ERROR.
                        0040    138 ;
    21 50    1F    E4    0040    139 CALERR: BBSC    #INTERROR,R0,10$        ; BR IF THIS IS A INTERNAL ERROR
```

```
                52   7C  0044   140          CLRQ    R2                              ; PRESET TOKEN DESCRIPTOR TO NULL
                55   D5  0046   141          TSTL    R5                              ; IS TOKEN INDEX VALID?
                1B   15  0048   142          BLEQ    10$                             ; IF NOT, THEN RETURN NULL DESCRIPTOR
            FFB3'    30  004A   143          BSBW    DCL$GETEXTDESC                  ; TAKE APART DESCRIPTOR(POINT OF ERROR)
                53   DD  004D   144          PUSHL   R3                              ; SAVE POINT OF ERROR
                55   D6  004F   145          INCL    R5                              ; ADVANCE TO NEXT
            FFAC'    30  0051   146          BSBW    DCL$GETEXTDESC                  ; TAKE THAT APART
          52    53   D0  0054   147          MOVL    R3,R2                           ; COPY ENDING ADDRESS OF ERROR
                08   BA  0057   148          POPR    #^M<R3>                         ; GET POINT OF ERROR BACK
          52    53   C2  0059   149          SUBL    R3,R2                           ; FIND LENGTH OF ERROR SEGMENT
                53   D7  005C   150          DECL    R3                              ; BACKUP TO PRECEEDING TERMAINATOR
          51    04   91  005E   151          CMPB    #PTR_K_ENDLINE,R1               ; WAS ERROR AT END-OF-LINE?
                02   12  0061   152          BNEQ    10$                             ; BR IF NO-ALL IS CORRECT
                52   D6  0063   153          INCL    R2                              ; ADJUST LENGTH FOR LAST LAST BYTE IN TOKEN
       08 A9    52   7D  0065   154 10$:     MOVQ    R2,CLI$Q_RQDESC(R9)             ; SET IN DESCRIPTOR
          04 A9 DD   0069   155          PUSHL   CLI$A_ERRACT(R9)                ; GET USER ERROR ROUTINE ADDRESS/OFFSET
                1B   13  006C   156          BEQL    RET0                            ; BR IF NO ERROR ROUTINE
       03 69    11   E0  006E   157          BBS     #CLI$V_ABSADR+<CLI$B_RQFLGS*8>,(R9),20$ ; BR IF ADR IS ABSOLUTE
                6E   59  C0  0072   158          ADDL    R9,(SP)                         ; FIND REAL ADDRESS OF ROUTINE
                5B   6A  D0  0075   159 20$:     MOVL    (R10),R11                       ; SET USER CONTEXT WORD
                5B   DD  0078   160          PUSHL   R11                             ; AND PASS IN ARGUMENT LIST
    50  00030000 8F   C8  007A   161          BISL    #<CLI$_ABKEYW&^X0FFF0000>,R0 ; INCLUDE SUBSYSTEM NUMBER
                50   DD  0081   162          PUSHL   R0                              ; ERROR CODE IS SECOND ARGUMENT INPUT,
                59   DD  0083   163          PUSHL   R9                              ; REQUEST DESCRIPTOR IS FIRST ARGUMENT
       0C BE    03   FB  0085   164          CALLS   #3,@12(SP)                      ; GIVE THE USER CHANCE TO HANDLE ERROR
                     04  0089   165 RET0:     RET                                     ; GO BACK FROM CALL BACK
                         008A   166
                         008A   167 ;
                         008A   168 ; RESULT PARSE DISPATCHER
                         008A   169 ;
                         008A   170
                   0000  008A   171 RSLTPRS:.WORD   0                               ; REGISTERS ALREADY SAVED!
    04    04       EF    008C   172          EXTZV   #CLI$V_PRITYP,#CLI$S_PRITYP,- ; EXTRACT THE PRIMARY REQUEST
    51    69             008F   173                  CLI$B_RQTYPE(R9),R1             ; FROM THE REQUEST DESCRIPTOR
          11   13  0091   174          BEQL    10$                             ; BR IF REQUEST IS UTILITY TYPE
    04    00       EF    0093   175          EXTZV   #CLI$V_SUBTYP,#CLI$S_SUBTYP,- ; GET THE PARAMETER NUMBER
    50    69             0096   176                  CLI$B_RQTYPE(R9),R0             ; OR SUB TYPE FOR RESULT
                         0098   177          CASE    R1,-                            ; DISPATCH ON REQUEST TYPE
                         0098   178                  LIMIT=#CLI$K_INPSPEC,<-         ;STARTING WTIH INPUT SPECIFICATION
                         0098   179                  SETQUAL,-                       ; REQUEST FOR INPUT DEFINTION
                         0098   180                  SETQUAL,-                       ; REQUEST FOR OUTPUT DEFINTION
                         0098   181                  CMPPRM,-                        ; COMPLETED WITH PARAMETER SET
                         0098   182                  DCL$VALCNV,-                    ; REQUEST FOR VALUE CONVERSION
                         0098   183                  >
                         00A4   184                                                  ; FALL THROUGH ON UTILITY OR ERROR
                         00A4   185 10$:     CASE    CLI$B_RQTYPE(R9),-              ; DECODE UTILITY REQUEST
                         00A4   186                  LIMIT=#CLI$K_INITPRS,-          ; LOW VALUE FOR CASE
                         00A4   187                  TYPE=B,<-                       ; TYPE OF CASE IS BYTE
                         00A4   188                  DCL$RPINIT,-                    ; INIT RESULT PARSE
                         00A4   189                  DCL$GETCMD,-                    ; GET COMMAND LINE DESCRIPTOR
                         00A4   190                  SETQUAL,-                       ; SET QUALIFER STATE
                         00A4   191                  DCL$GETOPT,-                    ; GET COMMAND OPTION
                         00A4   192                  DCL$GETLINE+2,-                 ; GET COMMAND LINE
                         00A4   193                  DCL$GETLINE+2,-                 ; ** CLISERV SPACE HOLDER **
                         00A4   194                  >
                         00B4   195          SETSTAT INVREQTYP                       ; INVALID REQUEST TYPE
                      04 00B9   196          RET                                     ; DONE WITH THIS COMMAND
```

```
                     00BA   197
                     00BA   198 ;
                     00BA   199 ; NOTE WHEN DISPATCHING  TO THESE BLISS ROUTINES, THE REGISTER MASK IS BEING
                     00BA   200 ; SKIPPED OVER.  THEREFORE, ALL REGISTERS MUST BE PUSHED BEFORE DISPATCHING.
                     00BA   201 ;
                     00BA   202
                     00BA   203 CLINT:  CASE    CLI$B_RQTYPE(R9),-          ; DECODE CLINT REQUEST
                     00BA   204                 LIMIT=#CLI$K_PRESENT,-      ; LOW VALUE FOR CASE
                     00BA   205                 TYPE=B,<-                   ; TYPE OF CASE IS BYTE
                     00BA   206                 DCL$PRESENT+2,-             ; DETERMINE IF ENTITY PRESENT
                     00BA   207                 DCL$GETVALUE+2,-            ; GET VALUE OF ENTITY
                     00BA   208                 DCL$ENDPARSE+2,-            ; CLEAN UP AFTER PARSING COMMAND LINE
                     00BA   209                 DCL$DCLPARSE+2,-            ; PARSE COMMAND LINE
                     00BA   210                 DCL$DISPATCH+2,-            ; DISPATCH TO ACTION ROUTINE
                     00BA   211                 DCL$NEXTQUAL+2,-            ; GET NEXT QUALIFIER
                     00BA   212                 >
              04     00CB   213         RET                                 ; DONE WITH THIS COMMAND
```

```
                     00CC    215          .SBTTL  ENDPRM CALLBACK
                     00CC    216 ;---
                     00CC    217 ;
                     00CC    218 ; FUNCTIONAL DESCRIPTION:
                     00CC    219 ;
                     00CC    220 ;       THIS ROUTINE IS CALLED WHEN ALL QUALIFIERS AND
                     00CC    221 ;       VALUES HAVE BEEN RETRIEVED FOR A GIVEN PARAMETER.
                     00CC    222 ;       A CHECK IS MADE TO ENSURE THAT ALL QUALIFIERS
                     00CC    223 ;       PRESENT ON THE COMMAND LINE HAVE BEEN PROCESSED
                     00CC    224 ;       BY THE UTILITY.
                     00CC    225 ;
                     00CC    226 ; INPUTS:
                     00CC    227 ;
                     00CC    228 ;       R0 = PARAMETER NUMBER TO BE TERMINATED
                     00CC    229 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR BLOCK
                     00CC    230 ;       R10 = ADDRESS OF IMAGE LOCAL WORK AREA
                     00CC    231 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     00CC    232 ;
                     00CC    233 ; OUTPUTS:
                     00CC    234 ;
                     00CC    235 ;       THE REQUEST IS PROCESSED.
                     00CC    236 ;---
                     00CC    237 CMPPRM:
 56   40 AA40   DE   00CC    238          MOVAL   RPW_G_PRMLIM(R10)[R0],R6 ; SET ADDRESS OF PROPER LIMIT DESC
      7E   01   7D   00D1    239          MOVQ    #1,-(SP)                 ; SET SUCCESS PLUS A ZERO LONG WORD
      58   5E   D0   00D4    240          MOVL    SP,R8                    ; MARK POINT OF ERROR PARAMETERS
                     00D7    241                                           ; NOTE: R5 WAS ZEROED IN INITIAL ENTRY
  08 AA    55   91   00D7    242 10$:     CMPB    R5,RPW_B_STRPARM(R10)    ; IS INDEX AT END COMD QUALIFIER AREA?
           06   12   00DB    243          BNEQ    20$                      ; BR IF NO
  55   01 A6   9A   00DD    244          MOVZBL  PLM_B_FSTDESC(R6),R5     ; ELSE SET START OF PARAMETER AREA
           60   13   00E1    245          BEQL    80$                      ; BR IF PARAMETER IS NON-EXISTANT
  02 A6    55   91   00E3    246 20$:     CMPB    R5,PLM_B_LSTDESC(R6)     ; IS INDEX OUT OF CURRENT PARAMETER?
           1A   1A   00E7    247          BGTRU   50$                      ; BR IF ALL DONE
        FF14'   30   00E9    248          BSBW    DCL$GETEXTDESC           ; GET AND EXTRACT DESCRITPTOR
                     00EC    249          ASSUME  PTR_K_PARMQUAL EQ 1      ;
                     00EC    250          ASSUME  PTR_K_COMDQUAL EQ 0      ;
     01    51   91   00EC    251          CMPB    R1,#PTR_K_PARMQUAL       ; ANY KIND OF QUALIFIER?
           0E   1A   00EF    252          BGTRU   40$                      ; IF NO BR AND CONTINUE SEARCH
09 20 AA    55   E0   00F1    253          BBS     R5,RPW_G_BITS(R10),40$   ; BR IF THE QUALIFIER HAS BEEN SEEN
                     00F6    254          SETSTAT UNPROQUAL
 46'AF    6C   FA   00FB    255          CALLG   (AP),B^100$              ; PROCESS ERROR CALL BACK
           55   D6   00FF    256 40$:     INCL    R5                       ; ADD 1 TO BUFFER INDEX
           D4   11   0101    257          BRB     10$                      ; KEEP LOOKING
     55    66   9A   0103    258 50$:     MOVZBL  PLM_B_NXTDESC(R6),R5     ; NEXT DESCRIPTOR TO PROCESS
           3B   13   0106    259          BEQL    80$                      ; BR IF NO PARAMETER PRESENT
  02 A6    55   91   0108    260          CMPB    R5,PLM_B_LSTDESC(R6)     ; ALL BEEN PROCESSED
           09   1A   010C    261          BGTRU   55$                      ; BR IF YES
                     010E    262          SETSTAT UNPROPARM                ; UNPROCESSED PARAMETERS
 46'AF    6C   FA   0113    263          CALLG   (AP),B^100$              ; GENERATE AN ERROR
 55   02 A6    9A   0117    264 55$:     MOVZBL  PLM_B_LSTDESC(R6),R5     ; INDEX TO LAST DESCRIPTOR
           55   96   011B    265          INCB    R5                       ; SET TO NEXT DESCRIPTOR INDEX
  03 A6    55   91   011D    266          CMPB    R5,PLM_B_TRMDESC(R6)     ; IS THIS THE TERMINATOR DESCRIPTOR?
           20   1E   0121    267          BGEQU   80$                      ; BR IF YES-NOTHING MORE TO DO!
     66    55   90   0123    268          MOVB    R5,PLM_B_NXTDESC(R6)     ; SET THAT AS NEXT FOR NEXT CALLBACK
        FED7'   30   0126    269 60$:     BSBW    DCL$GETPARM              ; FIND THE NEXT PARAMETER
     0A 50    E9   0129    270          BLBC    R0,70$                   ; BR IF NONE REMAIN
     01    53   91   012C    271          CMPB    R3,#PTR_K_BLANK          ; CHECK IF END OF PARAMETER LIST
```

```
                     05   13  012F   272          BEQL    70$                             ; BR IF YES-SET NEW LIMIT TO HERE
                05   53   91  0131   273          CMPB    R3,#PTR_K_COMMA                 ; HOW ABOUT LIST SEPARATOR?
                     F0   12  0134   274          BNEQ    60$                             ; KEEP LOOKING
          02 A6      55   01   83  0136   275 70$: SUBB3   #1,R5,PLM_B_LSTDESC(R6)        ; SET NEW LAST
                     04   88  013B   276          BISB    #CLI$M_MOREINP,-                ; SET FLAG THAT MORE INPUT EXISTS
                   03 A9        013D   277                  CLI$B_RQSTAT(R9)                ; INDICATE MORE DATA TO COME
                     66   90  013F   278          MOVB    PLM_B_NXTDESC(R6),-             ; SET PREVIOUS NEXT AS
                   01 A6        0141   279                  PLM_B_FSTDESC(R6)               ; AS FIRST IN THIS PARAMETER
                     21   BA  0143   280 80$:    POPR    #^M<R0,R5>                      ; GET STATUS AND POINT OF ERROR(IF ONE)
                     04   05  0145   281 90$:    RET
                          0146   282 ;
                          0146   283 ; THIS ROUTINE IS CALLED TO FACILITATE MULTIPLE ERROR
                          0146   284 ; ACTION CALLS WHEN PROCESSING THE END OF A PARAMETER SET
                          0146   285 ;
                   0820   0146   286 100$:  .WORD   ^M<R5,R11>                      ; SAVE REGISTERS 5 AND 11
                     21   BB  0148   287          PUSHR   #^M<R0,R5>                      ; SET ERROR AND PLACE IN THE LINE
                50   68   D0  014A   288          MOVL    (R8),R0                         ; GET PRVIOUS ERROR
             55   04   A8   D0  014D   289          MOVL    4(R8),R5                        ; GET PREVIOUS PLACE
                68   8E   7D  0151   290          MOVQ    (SP)+,(R8)                      ; SET THE NEW AS NEXT TO DO
                  EE 50   E8  0154   291          BLBS    R0,90$                          ; BR IF FIRST TIME THROUGH
                    FEE6   31  0157   292          BRW     CALERR                          ; GO CALL THE UTILITY WITH ERROR
```

```
                     015A   294          .SBTTL  INPUT(N),OUTPUT(N),GETQUAL CALLBACKS
                     015A   295 ;---
                     015A   296 ;
                     015A   297 ; FUNCTIONAL DESCRIPTION:
                     015A   298 ;
                     015A   299 ;       THIS ROUTINE HANDLES THE INPUT, OUTPUT AND GETQUAL
                     015A   300 ;       CALLBACKS TO SUPPLY AN INPUT/OUTPUT PARAMETER OR
                     015A   301 ;       PROCESS ALL QUALIFIERS ASSOCIATED WITH A GIVEN
                     015A   302 ;       PARAMETER OR VERB.
                     015A   303 ;
                     015A   304 ; INPUTS:
                     015A   305 ;
                     015A   306 ;       R0 = INPUT OR OUTPUT NUMBER (IF INPUT/OUTPUT REQUEST)
                     015A   307 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR BLOCK
                     015A   308 ;       R10 = ADDRESS OF IMAGE LOCAL WORK AREA
                     015A   309 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     015A   310 ;
                     015A   311 ; OUTPUTS:
                     015A   312 ;
                     015A   313 ;       THE REQUEST IS PROCESSED.
                     015A   314 ;---
                     015A   315 SETQUAL:
        58   0C AC   D0  015A   316          MOVL    RQBITS(AP),R8           ; GET USERS BIT ARRAY
                     015E   317 ;
                     015E   318 ; RESET ALL STATUS FLAGS AND DESCRIPTORS FOR ALL QUALIFIER BLOCKS
                     015E   319 ; LINKED TO THE CALLING REQUEST DESCRIPTOR BLOCK
                     015E   320 ;
    0244'CF          DF  015E   321          PUSHAL  W^SCANQUAL              ; SET INITIAL ADDRESS FOR QUALIFER SCAN
          9E         16  0162   322 10$:     JSB     @(SP)+                  ; CO-ROUTINE LINK TO SCAN QUALIFIERS
       08 50         E9  0164   323          BLBC    R0,20$                  ; BR WHEN ALL ARE SCANNED
       03 A7         94  0167   324          CLRB    CLI$B_QDSTAT(R7)        ; RESET ALL STATUS FLAGS
       04 A7         7C  016A   325          CLRQ    CLI$Q_QDVALDESC(R7)     ; SET VALUE DESCRIPTOR TO ZERO
          F3         11  016D   326          BRB     10$                     ; TRY FOR NEXT
                     016F   327 ;
                     016F   328 ; IF GETQUAL REQUEST, THEN FOR EACH QUALIFIER DESCRIPTOR BLOCK LINKED
                     016F   329 ; TO THIS REQUEST DESCRIPTOR, PROCESS THE COMMAND QUALIFIER (IF PRESENT).
                     016F   330 ;
    02   69         91  016F   331 20$:     CMPB    CLI$B_RQTYPE(R9),#CLI$K_GETQUAL ; IS THIS REQUEST FOR QUALIFER
            0A      12  0172   332          BNEQ    25$                     ; DEFINTION ONLY- BR IF NO
       C2 AB         90  0174   333          MOVB    WRK_B_VERBTYP(R11),-    ; SET COMMAND GENERIC VERB TYPE INTO
       03 A9                  0177   334                  CLI$B_RQSTAT(R9)        ; REQUEST DESCRIPTOR STATUS BYTE
        038E         30  0179   335          BSBW    DCL$PROCMDQUAL          ; FIND COMMAND QUALIFIER
          24         11  017C   336          BRB     40$                     ; TAKE ACTION
                     017E   337 ;
                     017E   338 ; IF INPUT(N) OR OUTPUT(N) REQUEST, THEN FIND THE PARAMETER OR QUALIFIER
                     017E   339 ; DESCRIBING THE INPUT OR OUTPUT AND PROCESS IT.
                     017E   340 ;
          00E7       30  017E   341 25$:     BSBW    PROCIO                  ; PROCESS INPUT/OUTPUT DESCRIPTION
    52    01 A9      9A  0181   342          MOVZBL  CLI$B_BITNUM(R9),R2     ; GET THE PARAMETER PRESENT FLAG BIT
          00         E0  0185   343          BBS     #CLI$V_PARMPRS,-        ; BR IF THE PARAMETER IS PRESENT
       14 03 A9              0187   344                  CLI$B_RQSTAT(R9),30$    ;
                     018A   345          CLRBIT  R2,(R8)                 ; INDICATE PARAMETER ABSENT
                     018E   346          SETSTAT REQPRMABS               ; SET REQUIRED PARAMETER ABSENT
          00         E0  0193   347          BBS     #CLI$V_PARMREQ,-        ; BR IF PARAMETER IS REQUIRED
       75 02 A9              0195   348                  CLI$B_RQFLGS(R9),140$   ;
    51    14 A9      D0  0198   349          MOVL    CLI$A_ABSACT(R9),R1     ; GET PARAMETER ABSENT ACTION ADDRESS
          65         11  019C   350          BRB     120$                    ; JOIN COMMON ROUTINE
```

```
                          019E    351 30$:    SETBIT  R2,(R8)                 ; SET PARAMETER PRESENT FLAG
                          01A2    352 ;
                          01A2    353 ; INITIALIZE 4 COROUTINE START ADDRESSES FOR THE FOLLOWING
                          01A2    354 ; 4 PASSES THROUGH ALL OF THE QUALIFIER DESCRIPTOR BLOCKS LINKED
                          01A2    355 ; TO THE REQUEST DESCRIPTOR.
                          01A2    356 ;
          0244'CF   9F    01A2    357 40$:    PUSHAB  W^SCANQUAL              ; SET INITIAL COROUTINE ADDRESS
                6E   DD   01A6    358         PUSHL   (SP)                    ; COPY COROUTINE INITIAL ADDRESS
                6E   DD   01A8    359         PUSHL   (SP)                    ; THREE MORE TIMES FOR
                6E   DD   01AA    360         PUSHL   (SP)                    ; SUBSEQUENT SCANS OF QUALIFIERS
                          01AC    361 ;
                          01AC    362 ; MARK ALL QUALIFIERS WITH DEFTRUE AS BEING PRESENT
                          01AC    363 ;
                9E   16   01AC    364 45$:    JSB     @(SP)+                  ; GET NEXT QUALIFIER DESCRIPTOR
              1A 50  E9   01AE    365         BLBC    R0,50$                  ; BR WHEN SCAN IS DONE
F6 03 A7        0?  E0   01B1    366         BBS     #CLI$V_QUALEXP,CLI$B_QDSTAT(R7),45$ ; LOOP IF FOUND EXPLICITLY
    51    01 A7   9A   01B6    367         MOVZBL  CLI$B_QDCODE(R7),R1     ; GET THE INDEX INTO THE TABLE
           FE43'  30   01BA    368         BSBW    DCL$GETPARMQUAL         ; FIND THE ASSOCIATED STRUCTURE
EA 04 A2     02  E1   01BD    369         BBC     #ENT_V_DEFTRUE,ENT_W_FLAGS(R2),45$ ; BR IF NOT DEFAULTED TRUE
    03 A7    01  88   01C2    370         BISB    #CLI$M_QUALTRU,CLI$B_QDSTAT(R7) ; MARK QUALIFIER TRUE
           03EE  30   01C6    371         BSBW    DCL$SETDEFVAL           ; SET UP THE DEFAULT VALUE IF THERE
             E1  11   01C9    372         BRB     45$                     ; LOOK AT NEXT
                          01CB    373 ;
                          01CB    374 ; FOR ALL QUALIFIERS NOT PRESENT, CLEAR THE ASSOCIATED BIT IN THE BIT MASK
                          01CB    375 ;
                9E   16   01CB    376 50$:    JSB     @(SP)+                  ; GET NEXT DESCRIPTOR
              0A 50  E9   01CD    377         BLBC    R0,60$                  ; BR WHEN NO MORE
F6 03 A7        00  E0   01D0    378         BBS     #CLI$V_QUALTRU,CLI$B_QDSTAT(R7),50$ ; BR IF TRUE
           02EF  30   01D5    379         BSBW    DCL$CLRSETLST           ; CLEAR THE BITS
             F1  11   01D8    380         BRB     50$                     ; LOOK FOR MORE FALSSE QUALIFIERS
                          01DA    381 ;
                          01DA    382 ; FOR ALL QUALIFIERS PRESENT, TEST/SET THE ASSOCIATED BIT IN THE BIT MASK
                          01DA    383 ;
                9E   16   01DA    384 60$:    JSB     @(SP)+                  ; GET NEXT QUALIFIER DESCRIPTOR
              13 50  E9   01DC    385         BLBC    R0,100$                 ; BR WHEN NO MORE
F6 03 A7        00  E1   01DF    386         BBC     #CLI$V_QUALTRU,CLI$B_QDSTAT(R7),60$ ; BR IF FALSE
             F3 AF  9F   01E4    387         PUSHAB  B^60$                   ; SUBROUTINE RETURN ADDRESS
03 03 A7        01  E1   01E7    388         BBC     #CLI$V_QUALEXP,CLI$B_QDSTAT(R7),70$ ; BR IF NOT EXPLICITLY FOUND
           02B5  31   01EC    389         BRW     DCL$TSTSETLST           ; TEST THEN SET SET LIST, ETC.
           02CC  31   01EF    390 70$:    BRW     DCL$SETSETLST           ; ONLY SET THE SET LIST FOR DEFAULTS
                          01F2    391 ;
                          01F2    392 ; FOR ALL QUALIFIERS, CALL THE ASSOCIATED ACTION ROUTINE (IF ANY)
                          01F2    393 ;
                9E   16   01F2    394 100$:   JSB     @(SP)+                  ; GET NEXT QUALIFIER DESCRIPTOR
              08 50  E9   01F4    395         BLBC    R0,110$                 ; BR WHEN NO MORE QUALIFIERS
                 10  E0   01F7    396         BBS     #CLI$V_ALLOCCUR+<CLI$B_QDFLGS*8>,- ; IF ALL OCCURANCES IS SET
              F7 67      01F9    397                 (R7),100$               ; CALL BACK ALREADY BEEN DONE
                 11  10   01FB    398         BSBB    QUALACT                 ; TAKE QUALIFIER ACTION
                 F3  11   01FD    399         BRB     100$                    ; TRY FOR NEXT
                          01FF    400 ;
                          01FF    401 ; CALL THE PARAMETER PRESENT/ABSENT ACTION ROUTINE (IF ANY)
                          01FF    402 ;
    51    10 A9   D0   01FF    403 110$:   MOVL    CLI$A_PRSACT(R9),R1     ; PRESENT ACTION ADDRESS OFFSET
                 05  13   0203    404 120$:   BEQL    130$                    ; BR IF NO PARAMETER ACTION
       50    59   D0   0205    405         MOVL    R9,R0                   ; SET ADDRESS OF ARGUMENT TO CALL WITH
                 20  10   0208    406         BSBB    CALLBAK                 ; ISSUE CALL BACK
                          020A    407 130$:   SETSTAT SUCCESS                 ; SET GOOD RETURN
```

```
                04  020D   408 140$:   RET                         ; BACK TO DISPATCHER
```

```
                                020E    410          .SBTTL  ACTION CALLBACK SUBROUTINE
                                020E    411 ;---
                                020E    412 ;
                                020E    413 ; FUNCTIONAL DESCRIPTION:
                                020E    414 ;
                                020E    415 ;       CALL THE USER'S ACTION ROUTINE IF SPECIFIED.
                                020E    416 ;
                                020E    417 ; INPUTS:
                                020E    418 ;
                                020E    419 ;       R7 = ADDRESS OF QUALIFIER DESCRIPTOR BLOCK
                                020E    420 ;       R10 = ADDRESS OF IMAGE LOCAL WORK AREA
                                020E    421 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                                020E    422 ;
                                020E    423 ;---
                                020E    424          .ENABL  LSB
                                020E    425
05 02 A7    02    E1            020E    426 QUALACT:BBC     #CLI$V_QDEXPA,CLI$B_QDFLGS(R7),5$ ; BR IF ACTION ALWAYS DESIRED
2B 03 A7    01    E1            0213    427         BBC     #CLI$V_QUALEXP,CLI$B_QDSTAT(R7),40$ ; IF NOT EXPLICIT
   51    10 A7    D0            0218    428 5$:     MOVL    CLI$A_FLSACT(R7),R1     ; ASSUME QUALIFIER IS FALSE
             00    E1            021C    429         BBC     #CLI$V_QUALTRU,-        ; BR IF THAT ASSUMPTION
       04 03 A7                  021E    430                 CLI$B_QDSTAT(R7),10$    ; WAS CORRECT
   51    0C A7    D0            0221    431         MOVL    CLI$A_TRUACT(R7),R1     ; GET TRUE ACTION ADDRESS OFFSET
             1C    13            0225    432 10$:    BEQL    40$                     ; BR IF NO ACTION ROUTINE
      50    57    D0            0227    433         MOVL    R7,R0                   ; ARGUMENT FOR CALL BACK
                                022A    434
                                022A    435 ;
                                022A    436 ; ENTER HERE WITH R0 SET TO ACTION ROUTINE ADDRESS
                                022A    437 ;
                                022A    438
03 69    11    E0               022A    439 CALLBAK:BBS     #CLI$V_ABSADR+<CLI$B_RQFLGS*8>,(R9),20$ ; BR IF ADR IS ABSOLUTE
   51    50    C0               022E    440         ADDL    R0,R1                   ; RELOCATE ADDRESS
   5B    6A    D0               0231    441 20$:    MOVL    (R10),R11               ; SET USER CONTEXT WORD
         5B    DD               0234    442         PUSHL   R11                     ; PASS USER CONTEXT WORD
   FDC6  CF    9F               0236    443         PUSHAB  DCL$UTLSERV             ; GIVE THE ACTION ROUTINE CALL BACK ADR
         60    9F               023A    444         PUSHAB  (R0)                    ; PASS CALLERS STRUCTURE AS ARGUMENT
   61    03    FB               023C    445         CALLS   #3,(R1)                 ; CALL THE ACTION ROUTINE
5B   04 AA    D0               023F    446         MOVL    RPW_L_DCLWRK(R10),R11   ; RESET THE COMMAND WORK ADDRESS
               05               0243    447 40$:    RSB                             ; RETURN TO MY CALLER
                                0244    448
                                0244    449          .DSABL  LSB
```

```
                     0244   451          .SBTTL  SCAN QUALIFIER DESCRIPTOR LIST
                     0244   452 ;---
                     0244   453 ;
                     0244   454 ; FUNCTIONAL DESCRIPTION:
                     0244   455 ;
                     0244   456 ;       THIS CO-ROUTINE IS USED TO SCAN THE UTILITY'S QUALIFIER
                     0244   457 ;       DESCRIPTOR BLOCKS LINKED TO THE CURRENT REQUEST DESCRIPTOR.
                     0244   458 ;       THE CALLER IS CALLED BACK ONCE FOR EACH QUALIFIER DESCRIPTOR
                     0244   459 ;       BLOCK UNTIL R0 IS RETURNED FALSE.
                     0244   460 ;
                     0244   461 ; INPUTS:
                     0244   462 ;
                     0244   463 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR BLOCK
                     0244   464 ;
                     0244   465 ; OUTPUTS:
                     0244   466 ;
                     0244   467 ;       R7 = ADDRESS OF QUALIFIER DESCRIPTOR BLOCK
                     0244   468 ;       R0 = TRUE IF STILL MORE TO GO,
                     0244   469 ;            FALSE IF NO MORE LEFT
                     0244   470 ;---
                     0244   471 SCANQUAL:                                    ; SCAN QUALIFIERS
57   18 A9    D0     0244   472          MOVL    CLI$A_QUALST(R9),R7         ; GET OFFSET TO QUALIFIER LIST
       18    13     0248   473          BEQL    20$                         ; BR IF NONE AT ALL
03 69    11    E0   024A   474          BBS     #CLI$V_ABSADR+<CLI$B_RQFLGS*8>,(R9),10$ ; BR IF ADR IS ABSOLUTE
   57    59    C0   024E   475          ADDL    R9,R7                       ; ADJUST ADDRESS TO ABSOLUTE
                     0251   476 10$:    SETSTAT NORMAL                      ; ASSUME MORE QUALIFIERS TO PROCESS
         67    95   0254   477          TSTB    (R7)                        ; END OF LIST
         0A    13   0256   478          BEQL    20$                         ; BR IF END OF LIST
         9E    16   0258   479          JSB     @(SP)+                      ; RETURN WITH A DESCRIPTOR
   50    67    9A   025A   480          MOVZBL  CLI$B_QDBLKSIZ(R7),R0       ; GET SIZE OF DESCRIPTOR
   57    50    C0   025D   481          ADDL    R0,R7                       ; ADVANCE TO NEXT BLOCK
         EF    11   0260   482          BRB     10$                         ; CK IF MORE
                     0262   483 20$:    SETSTAT INVQUAL                     ; RETURN AN ERROR
               05    0267   484 RSB0:    RSB                                ; RETURN TO CALLER
```

```
                     0268   486          .SBTTL  PROCESS AN INPUT/OUTPUT REQUEST
                     0268   487 ;---
                     0268   488 ;
                     0268   489 ; FUNCTIONAL DESCRIPTION:
                     0268   490 ;
                     0268   491 ;       THIS ROUTINE IS CALLED TO PROCESS A GIVEN INPUT OR
                     0268   492 ;       OUTPUT FOR THE UTILITY.  THE INPUT OR OUTPUT MAY BE
                     0268   493 ;       SPECIFIED EITHER BY A PARAMETER OR QUALIFIER, DEPENDING
                     0268   494 ;       ON THE COMMAND DEFINITION.
                     0268   495 ;
                     0268   496 ; INPUTS:
                     0268   497 ;
                     0268   498 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR BLOCK
                     0268   499 ;       R10 = ADDRESS OF IMAGE LOCAL WORK AREA
                     0268   500 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     0268   501 ;
                     0268   502 ; OUTPUTS:
                     0268   503 ;
                     0268   504 ;       PARMPRS BIT IS SET IF INPUT/OUTPUT IS PRESENT.
                     0268   505 ;       QUADWORD DESCRIPTOR DESCRIBES INPUT/OUTPUT SPECIFICATION.
                     0268   506 ;---
                     0268   507 PROCIO:
04   00   EF          0268   508          EXTZV   #CLI$V_SUBTYP,#CLI$S_SUBTYP,- ; AND THE SUB TYPE VIELD
50   69               026B   509                  CLI$B_RQTYPE(R9),R0     ; INTO R0
04   04   ED          026D   510          CMPZV   #CLI$V_PRITYP,#CLI$S_PRITYP,-; CHECK THE PRIMARY REQUEST TYPE TO
02   69               0270   511                  CLI$B_RQTYPE(R9),#CLI$K_OUTSPEC ; SEE IF REQUEST IS FOR OUTPUT
     03   13          0272   512          BEQL    OUTPUT                  ; BR IF REQUEST IS FOR OUTPUT
  0121    31          0274   513          BRW     INPUT                   ; ELSE PROCESS INPUT
                     0277   514 ;
                     0277   515 ; PROCESS REQUEST FOR AN OUTPUT SPECIFICATION
                     0277   516 ;
                     0277   517 OUTPUT:                                   ; REQUEST ID FOR OUTPUT SPEC
51   D2 AB   D0      0277   518          MOVL    WRK_L_PAROUT(R11),R1    ; SET POINTER TO OUTPUT PARSE TABLE
     EA   13         027B   519          BEQL    RSB0                    ; BR IF NO TABLE
   81   50   91      027D   520          CMPB    R0,(R1)+                ; REQUEST IN RANGE?
     E5   1E         0280   521          BGEQU   RSB0                    ; BR IF NO
                     0282   522 ;
                     0282   523 ; IF THE OUTPUT PARSE INDICATOR IS NEGATIVE, THEN SIMPLY USE
                     0282   524 ; IT AS THE NEGATED PARAMETER NUMBER BY INDEXING INTO THE PARAMETER
                     0282   525 ; LIMIT TABLE.
                     0282   526 ;
 51   6140   98      0282   527          CVTBL   (R1)[R0],R1             ; GET OUTPUT PARSE INDICATOR
F8 8F   51   91      0286   528          CMPB    R1,#-CMD_K_MAX_PARMS    ; PARAMETER OR QUALIFIER?
     03   1B         028A   529          BLEQU   10$                     ; BR IF OUTPUT IS DEFINED BY QUALIFIER
   0106   31         028C   530          BRW     95$                     ; ELSE IT IS A FORMAL PARAMETER
                     028F   531 ;
                     028F   532 ; LOCATE THE QUALIFIER DESCRIPTOR WHICH DESCRIBES THIS OUTPUT
                     028F   533 ;
     FD6E'   30      028F   534 10$:     BSBW    DCL$GETQUALDESC         ; FIND DESCRIPTOR FOR QUALIFIER #(R1)
                     0292   535 ;
                     0292   536 ; IF THE QUALIFIER IS DEFAULTED TRUE, SET THE OUTPUT PRESENT AND DEFAULTED.
                     0292   537 ; NOTE THAT THE PARMPRS AND PARMDEF FLAGS HAVE ALREADY BEEN PRESET FALSE.
                     0292   538 ;
18 04 A2   02   E0   0292   539          BBS     #ENT_V_DEFTRUE,ENT_W_FLAGS(R2),25$ ; BR IF DEFAULTED TRUE
17 04 A2   03   E1   0297   540          BBC     #ENT_V_BATDEF,ENT_W_FLAGS(R2),30$ ; BR IF NOT BATCH DEFAULTED
     5B   DD        029C   541          PUSHL   R11                     ; SAVE WRK ADDRESS
  00000000'EF   16   029E   542          JSB     CLI$GET_PRC             ; GET ADDRESS OF PRC IN R11
```

```
                50    5B    D0  02A4   543          MOVL    R11,R0                  ; MOVE INTO R0
                      5B  8ED0  02A7   544          POPL    R11                     ; RESTORE WRK ADDRESS
          04 68 A0    06    E1  02AA   545          BBC     #PRC_V_MODE,PRC_W_FLAGS(R0),30$ ;BRANCH IF NOT BATCH JOB
                      09    88  02AF   546 25$:      BISB    #CLI$M_PARMPRS!CLI$M_PARMDEF,-; SET PARAMETER PRESENT & DEFAULT
                   03 A9          02B1   547                  CLI$B_RQSTAT(R9)        ; IN REQUEST STATUS BYTE
                                  02B?   548 ;
                                  02B3   549 ; IF THERE IS A DEFAULT VALUE ASSOCIATED WITH THIS QUALIFIER, THEN
                                  02B3   550 ; RETURN ITS DESCRIPTOR IN THE REQUEST DESCRIPTOR BLOCK.  OF COURSE,
                                  02B3   551 ; THIS IMPLIES THAT THE VALUE DESCRIPTOR SHOULD ONLY BE USED IF THE
                                  02B3   552 ; PARMPRS BIT IS SET SINCE THE VALUE WILL ALWAYS BE THERE EVEN THOUGH
                                  02B3   553 ; THE QUALIFIER IS NOT.
                                  02B3   554 ;
             50    1C A2    32    02B3   555 30$:      CVTWL   ENT_W_DEFVAL(R2),R0     ; GET OFFSET TO DEFAULT VALUE
                      0E    13    02B7   556          BEQL    35$                     ; BRANCH IF NONE
                50    01    C0    02B9   557          ADDL    #1,R0                   ; CALCULATE ADDRESS OF COUNTED STRING
                50    52    C0    02BC   558          ADDL    R2,R0                   ;
             08 A9    80    9A    02BF   559          MOVZBL  (R0)+,CLI$Q_RQDESC(R9)  ; STORE LENGTH INTO VALUE DESCRIPTOR
             0C A9    50    D0    02C3   560          MOVL    R0,CLI$Q_RQDESC+4(R9)   ; AND ADDRESS
                                  02C7   561 ;
                                  02C7   562 ; LOCATE THE LAST OCCURRENCE OF THE QUALIFIER ON THE COMMAND LINE
                                  02C7   563 ;
                      7E    D4    02C7   564 35$:      CLRL    -(SP)                   ; MAKE SPACE FOR PARAMETER LIMIT DESC
                      7E    7C    02C9   565          CLRQ    -(SP)                   ; SET VALUES FOR QUALIFER TO ZERO
                0000'CF    9F    02CB   566          PUSHAB  W^DCL$FNDCMDQUAL        ; SET COROUTINE ADDRESS
                      9E    16    02CF   567 40$:      JSB     @(SP)+                  ; COROUTINE LINK
                   15 50    E9    02D1   568          BLBC    R0,50$                  ; BR IF NO MORE COMMADN QUALIFIERS
             51    05 A4    91    02D4   569          CMPB    PTR_B_NUMBER(R4),R1     ; IS THIS THE QUALIFIER FOR THIS OUTPUT?
                      F5    12    02D8   570          BNEQ    40$                     ; BR IF NO
             04 AE    54    7D    02DA   571          MOVQ    R4,4(SP)                ; SAVE DESCRIPTOR ADDRESS AND INDEX
             0C AE    56    D0    02DE   572          MOVL    R6,12(SP)               ; SAVE PARAMETER LIMIT DESCRIPTOR
                                  02E2   573          SETBIT  R5,RPW_G_BITS(R10)      ; INDICATE THAT QUALIFIER WAS USED
                      E6    11    02E7   574          BRB     40$                     ; LOOK FOR ANOTHER OCCURANCE
                                  02E9   575 ;
                                  02E9   576 ; SET THE PARMPRS AND PARMDEF FLAGS DEPENDING ON WHETHER THE
                                  02E9   577 ; QUALIFIER WAS FOUND AND WHETHER IT IS NEGATED.
                                  02E9   578 ;
                54    8E    7D    02E9   579 50$:      MOVQ    (SP)+,R4                ; RETREIVE PARAMETERS FOR LAST OCCURANCE
                56    8E    D0    02EC   580          MOVL    (SP)+,R6                ; RESET PARAMETER LIMIT DESCRIPTOR
                      44    13    02EF   581          BEQL    80$                     ; BR IF NOT IN COMMAND EXPLICITLY
                      09    8A    02F1   582          BICB    #CLI$M_PARMPRS!CLI$M_PARMDEF,-; CLR PARAMETER PRESENT & DEFAULT
                   03 A9          02F3   583                  CLI$B_RQSTAT(R9)        ; IN REQUEST STATUS BYTE
          3C 64    14    E0    02F5   584          BBS     #PTR_V_NEGATE,(R4),80$ ; BR IF ASSUMED CORRECTLY
          03 A9    01    88    02F9   585          BISB    #CLI$M_PARMPRS,CLI$B_RQSTAT(R9) ; SET EXPLICITLY PRESENT
                                  02FD   586 ;
                                  02FD   587 ; IF THERE IS A VALUE ON THE QUALIFIER, USE THAT VALUE
                                  02FD   588 ;
                54    0C    C0    02FD   589          ADDL    #PTR_C_LENGTH,R4        ; ADVANCE POINTER TO NEXT DESCRIPTOR
                   FCFD'    30    0300   590          BSBW    DCL$EXTRSLDESC          ; TAKE DESCRIPTOR APART
                02    51    91    0303   591          CMPB    R1,#PTR_K_QUALVALU      ; IS THIS A QUALIFIER VALUE?
                      29    13    0306   592          BEQL    70$                     ; BR IF FILENAME HERE AS QUALIFIER VALUE
                                  0308   593 ;
                                  0308   594 ; USE THE FILE SPECIFICATION ON THE PARAMETER FOR THIS QUALIFIER
                                  0308   595 ; REMOVING THE FILE TYPE AND VERSION
                                  0308   596 ;
                      55    D7    0308   597 60$:      DECL    R5                      ; BACKUP IN RESULT PARSE DECSRIPTOR
                      29    15    030A   598          BLEQ    80$                     ; BRANCH IF NO PREVIOUS PARAMETERS
          01 A6    55    91    030C   599          CMPB    R5,PLM_B_FSTDESC(R6)    ; IS THIS IN THE CURRENT PARAMETER
```

```
                23   19  0310   600          BLSS    80$                     ; BR IF NO - NO MORE TO CHECK
              FCEB'  30  0312   601          BSBW    DCL$GETEXTDESC          ; TAKE THAT DESCRIPTOR APART
          03    51   91  0315   602          CMPB    R1,#PTR_K_PARAMETR      ; WAS THIS A PARAMETER?
                EE   12  0318   603          BNEQ    60$                     ; BR IF NO
    63    52    5D 8F   3A  031A   604          LOCC    #^A/]/,R2,(R3)          ; IS THERE A DIRECTORY SPEC HERE
                09   12  031F   605          BNEQ    65$                     ; BR IF YES
      63    52    3E   3A  0321   606          LOCC    #^A/>/,R2,(R3)          ; CHECK FOR ALTERNATE SYNTAX
                03   12  0325   607          BNEQ    65$                     ; BR IF THAT TYPE IS HERE
          50    52   7D  0327   608          MOVQ    R2,R0                   ; SET LIMITS FOR SEARCH FOR TYPE
    61    50    2E   3A  032A   609 65$:     LOCC    #^A/./,R0,(R1)          ; TRY TO FIND TYPE FIELD
          52    50   C2  032E   610          SUBL    R0,R2                   ; ADJUST LENGTH FOR FILE TYPE
       08 A9    52   7D  0331   611 70$:     MOVQ    R2,CLI$Q_RQDESC(R9)     ; SET RETURNED VALUE FOR NAME
                             0335   612 ;
                             0335   613 ; IF THE INPUT/OUTPUT WAS FOUND, THEN PROCESS ALL QUALIFIERS
                             0335   614 ; ASSOCIATED WITH THE PARAMETER ON WHICH IT WAS FOUND.
                             0335   615 ;
 5A 03 A9    00   E1  0335   616 80$:     BBC     #CLI$V_PARMPRS,CLI$B_RQSTAT(R9),90$ ; BR IF PARAMETER NOT HERE
             01CD  30  033A   617          BSBW    DCL$PROCMDQUAL          ; PROCESS COMMAND QUALIFIERS
                             033D   618 ;
                             033D   619 ; IF THE STRING DESCRIPTOR IS STILL NOT BEEN FILLED DESCRIBING
                             033D   620 ; THE OUTPUT SPECIFICATION, THEN TAKE THE PARAMETER MINUS THE
                             033D   621 ; FILE TYPE AND VERSION AND USE IT.
                             033D   622 ;
 52 02 A9    02   E0  033D   623          BBS     #CLI$V_EXPNAM,CLI$B_RQFLGS(R9),90$ ; BR IF EXPLICIT NAMES ONLY
          08 A9    B5  0342   624          TSTW    CLI$W_RQSIZE(R9)        ; NAME FOUND FOR THIS PARAMETER?
                4D   12  0345   625          BNEQ    90$                     ; BR IF YES
       56   40 AA   DE  0347   626          MOVAL   RPW_G_PRMLIM(R10),R6    ; POINT AT FIRST LIMIT DESCRIPTOR
       55   01 A6   9A  034B   627          MOVZBL  PLM_B_FSTDESC(R6),R5    ; INDEX TO FIRST PARAMETER
                43   13  034F   628          BEQL    90$                     ; BRANCH IF NO PARAMETER PRESENT
              FCAC'  30  0351   629          BSBW    DCL$SETDESCADR          ; SET ADDRESS OF DESCRIPTOR IN R4
              FCA9'  30  0354   630          BSBW    DCL$EXTRSLDESC          ; TAKE THE DESCRIPTOR A PART
    63    52    5D 8F   3A  0357   631          LOCC    #^A/]/,R2,(R3)          ; LOOK FOR A DIRECTORY SPEC
                1D   12  035C   632          BNEQ    84$                     ; BR IF FOUND A DIRECTORY
      63    52    3E   3A  035E   633          LOCC    #^A/>/,R2,(R3)          ; IF NO LOOK FOR THE OTHER DIRECTORY END
                17   12  0362   634          BNEQ    84$                     ; BR IF THAT DIRECTORY WAS FOUND
      63    52    3A   3A  0364   635          LOCC    #^A/:/,R2,(R3)          ; NOW LOOK FOR DEVICE NAME
                19   13  0368   636          BEQL    86$                     ; BR IF NO DEVICE NAME HERE
                7E   7C  036A   637          CLRQ    -(SP)                   ; MAKE A QUADWORD BUFFER
          6E    50   7D  036C   638 82$:     MOVQ    R0,(SP)                 ; SAVE LAST COLON WAS FOUND
                51   D6  036F   639          INCL    R1                      ; ADVANCE ADDRESS OVER THAT COLON
                50   D7  0371   640          DECL    R0                      ; SUBTRACT 1 FROM COUNT FOR THE COLON
    61    50    3A   3A  0373   641          LOCC    #^A/:/,R0,(R1)          ; LOOK FOR MORE COLONS
                F3   12  0377   642          BNEQ    82$                     ; BR IF MORE COLONS HERE
                03   BA  0379   643          POPR    #^M<R0,R1>              ; GET ADDRESS OF LENGTH FOR LAST COLON
       52    FF A0   9E  037B   644 84$:     MOVAB   -1(R0),R2               ; SET LENGTH MINUS THE TERMINATOR
       53    01 A1   9E  037F   645          MOVAB   1(R1),R3                ; AND ADDRESS BEYOND THE TERMINATOR
    63    52    2E   3A  0383   646 86$:     LOCC    #^A/./,R2,(R3)          ; LOOK FOR A TYPE FIELD
                04   12  0387   647          BNEQ    88$                     ; BR IF TYPE FIELD PRESENT
    63    52    3B   3A  0389   648          LOCC    #^A/;/,R2,(R3)          ; NOW LOOK FOR EXPLICIT VERSION
          52    50   C2  038D   649 88$:     SUBL    R0,R2                   ; ALSO REMOVE THAT IF FOUND
       08 A9    52   7D  0390   650          MOVQ    R2,CLI$Q_RQDESC(R9)     ; SET SIZE AND ADDRESS OF DESCRIPTOR
                05   05  0394   651 90$:     RSB                             ; RETURN TO DISPATCHER
                             0395   652
                             0395   653 ;
                             0395   654 ; COME HERE WHEN OUTPUT IS DEFINED BY A NEGATED PARAMETER NUMBER
                             0395   655 ;
                             0395   656
```

```
                     50   51    D2  0395   657 95$:    MCOML   R1,R0                    ; GET POSITIVE OUTPUT NUMBER - 1
                                  0398   658                                           ; BASED AT ZERO (P1=0)
                                  0398   659
                                  0398   660 ;
                                  0398   661 ; COME HERE WHEN INPUT IS REQUESTED (MUST BE A PARAMETER)
                                  0398   662 ;
                                  0398   663
                56   40 AA40    DE   0398   664 INPUT:  MOVAL   RPW_G_PRMLIM(R10)[R0],R6; GET ADDRESS OF PARAMETER LIMIT ENTRY
                                  039D   665
                                  039D   666 ;
                                  039D   667 ; RETURN THE VALUE OF THE PARAMETER DESCRIBED BY THE PARAMETER
                                  039D   668 ; LIMIT MARKER POINTED TO BY R6.  ALSO, PROCESS ALL QUALIFIERS
                                  039D   669 ; ASSOCIATED WITH THE PARAMETER.
                                  039D   670 ;
                                  039D   671
                    FC60'   30    039D   672          BSBW    DCL$EXTNXTDESC          ; TAKE NEXT DESCRIPTOR APART
                  43   50   E9    03A0   673          BLBC    R0,90$                  ; ALL DONE
            08 A9      52   7D    03A3   674          MOVQ    R2,CLI$Q_RQDESC(R9)     ; SAVE ADDRESS OF PARAMETER
                       01   88    03A7   675          BISB    #CLI$M_PARMPRS,-        ; SET FLAG THAT PARAMETER IS PRESENT
                  03 A9           03A9   676                  CLI$B_RQSTAT(R9)        ; IN USERS REQUEST STATUS BYTE
                       56   DD    03AB   677          PUSHL   R6                      ; SAVE PARAMETER DESCRIPTOR POINTER
                    015A   30    03AD   678          BSBW    DCL$PROCMDQUAL          ; TAKE CARE OF COMMAND QUALIFIERS
                       56 8ED0    03B0   679          POPL    R6                      ; RESTORE PARM LIMIT DESCRITPTOR POINTER
                    FC4A'  30    03B3   680 10$:     BSBW    DCL$EXTNXTDESC          ; TAKE THE NEXT DESCRIPTOR APART
                  2D   50   E9    03B6   681          BLBC    R0,90$                  ; BR IF NO MORE
                01    51   91    03B9   682          CMPB    R1,#PTR_K_PARMQUAL      ; IS THIS A PARAMETER QUALIFIER
                       17   12    03BC   683          BNEQ    30$                     ; BR IF NO
                FE82   CF   9F    03BE   684          PUSHAB  SCANQUAL                ; SET INITIAL COROUTINE ADDRESS
                       9E   16    03C2   685 20$:     JSB     @(SP)+                  ; GET NEXT DESCRIPTOR
                  20   50   E9    03C4   686          BLBC    R0,RET1                 ; NO FIND IS AN ERROR
                  05   A4   91    03C7   687          CMPB    PTR_B_NUMBER(R4),-      ; IS THIS THE QUALIFIER DESCRIPTOR?
                  01   A7         03CA   688                  CLI$B_QDCODE(R7)        ;
                       F4   12    03CC   689          BNEQ    20$                     ; IF NO LOOK AT NEXT
                       8E   D5    03CE   690          TSTL    (SP)+                   ; CLEAR COROUTINE ADDRESS
                    0187   30    03D0   691          BSBW    DCL$HANDLQUAL           ; SET UP QUALIFER RESULT PARSE DATA
                       DE   11    03D3   692          BRB     10$                     ; CHECK FOR MORE
             03        51   91    03D5   693 30$:     CMPB    R1,#PTR_K_PARAMETR      ; THE NEXT PARAMETER
                       D9   12    03D8   694          BNEQ    10$                     ; BR IF NO
                       66   97    03DA   695          DECB    PLM_B_NXTDESC(R6)       ; BACK UP INDEX FOR NEXT RESULT PARSE
                       66   91    03DC   696          CMPB    PLM_B_NXTDESC(R6),-     ; CHECK IF NEXT IS LEQ LAST,
                  02   A6         03DE   697                  PLM_B_LSTDESC(R6)       ; IN THE CURRENT PARAMETER
                       04   1A    03E0   698          BGTRU   90$                     ; IF GTRU, NO MORE ELEMEMTS IN THIS SET
                       02   88    03E2   699          BISB    #CLI$M_CONCATINP,-      ; SET FLAG TO SAY CONCATONATED INPUT
                  03   A9         03E4   700                  CLI$B_RQSTAT(R9)        ; LIST IS NO EXHAUSTED.
                            05    03E6   701 90$:     RSB                             ; BACK TO I/O PROCESSOR
                            04    03E7   702 RET1:    RET
```

```
                     03E8   704          .SBTTL  VALUE CONVERSION ROUTINES
                     03E8   705 ;++
                     03E8   706 ; FUNCTIONAL DESCRIPTION:
                     03E8   707 ;
                     03E8   708 ;       THIS ROUTINE IS CALLED WHEN THE UTILITY HAS REQUESTED
                     03E8   709 ;       A QUALIFIER VALUE CONVERSION.
                     03E8   710 ;
                     03E8   711 ; CALLING SEQUENCE:
                     03E8   712 ;
                     03E8   713 ;       ENTERED VIA A CASE FOLLOWING A CALL
                     03E8   714 ;
                     03E8   715 ; INPUT PARAMETERS:
                     03E8   716 ;
                     03E8   717 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR FOR VALUE CONVERSION
                     03E8   718 ;       R10 = ADDRESS OF IMAGE LOCAL WORK AREA
                     03E8   719 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     03E8   720 ;
                     03E8   721 ; OUTPUT PARAMETERS:
                     03E8   722 ;
                     03E8   723 ;       VALUE IS CONVERTED AND STRING DESCRIPTOR IN QUALFIER DESCRIPTOR
                     03E8   724 ;       IS UPDATED TO DESCRIBE THE REMAINING VALUE-IF ANY.
                     03E8   725 ;
                     03E8   726 ; COMPLETION CODES:
                     03E8   727 ;
                     03E8   728 ;       DCL$NORMAL FOR SUCCESSFUL CONVERSION
                     03E8   729 ;       DCL$VALCNVERR FOR CONVERSION ERROR
                     03E8   730 ;       DCL$NOVALUE IF VALUE NOT PRESENT
                     03E8   731 ;
                     03E8   732 ;--
                     03E8   733
                     03E8   734 DCL$VALCNV:                                  ; REQUEST FOR VALUE CONVERSION
          57   0C AC  D0  03E8   735          MOVL    12(AP),R7               ; GET QUALIFIER DESCRIPTOR
                     03EC   736          SETINTR NOVALUE                 ; ASSUME NO VALUE PRESENT
          52   04 A7  7D  03F3   737          MOVQ    CLI$Q_QDVALDESC(R7),R2  ; COPY QUALIFIER VALUE STRING DESCRITOR
               59  13  03F7   738          BEQL    40$                     ; BR IF NO VALUE
               7E  7C  03F9   739          CLRQ    -(SP)                   ; ASSUME NOT CONVERTING DEFAULT VALUE
          55   53  D0  03FB   740          MOVL    R3,R5                   ; COPY ADDRESS OF STRING
     54   F492 CB  DE  03FE   741          MOVAL   WRK_G_BUFFER(R11),R4    ; BASE ADDRESS OF BUFFER
          55   54  C2  0403   742          SUBL    R4,R5                   ; FIND BYTE OFFSET INTO BUFFER
     54   F9B6 CB  DE  0406   743          MOVAL   WRK_G_RESULT(R11),R4    ; RESULT PARSE BUFFER
     51     F4 A4  DE  040B   744          MOVAL   -PTR_C_LENGTH(R4),R1    ; SET INDEX BASE INTO RESULT BUFFER
          0C   08  ED  040F   745 10$:     CMPZV   #PTR_V_OFFSET,#PTR_S_OFFSET,-
          55   64          0412   746                  (R4),R5                 ; IS THIS THE DESCRIPTOR?
               0C  12  0414   747          BNEQ    20$                     ; BR IF FOUND THE DESCRIPTOR
     55   54   51  C3  0416   748          SUBL3   R1,R4,R5                ; FIND BYTE OFFSET TO DESCRIPTOR
          55   0C  C6  041A   749          DIVL    #PTR_C_LENGTH,R5        ; NOW PTR INDEX INTO RESULT BUFFER
             FBE0' 30  041D   750          BSBW    DCL$EXTRSLDESC          ; TAKE RESULT DESCRIPTOR APART
               22  11  0420   751          BRB     30$                     ; PROCESS THE REQUEST WITH USER VALUE
          54   0C  C0  0422   752 20$:     ADDL    #PTR_C_LENGTH,R4        ; ADVANCE TO NEXT DESCRIPTOR
     B6   AB   54  D1  0425   753          CMPL    R4,WRK_L_RSLEND(R11)    ; IS THIS THE LAST DESCRIPTOR?
               E4  1F  0429   754          BLSSU   10$                     ; BR IF NO
                   042B   755          SETBIT  #31,(SP)                ; SET FLAG FOR INTERNAL VALUE
     63   52   2C  3A  042F   756          LOCC    #^A/,/,R2,(R3)          ; CHECK FOR MULTIPLE VALUES
               0F  13  0433   757          BEQL    30$                     ; BR IF LAST VALUE VALUES
          52   50  C2  0435   758          SUBL    R0,R2                   ; FIND LENGTH OF CURRENT VALUE
     6E   50   01  A3  0438   759          SUBW3   #1,R0,(SP)              ; SET REMAINING LENTH
04   AE   53   52  C1  043C   760          ADDL3   R2,R3,4(SP)             ; FIND ADDRESS OF COMMA
```

```
                04 AE    D6  0441   761                 INCL    4(SP)                           ; PLUS 1 TO START OF REAL VALUE
                            0444   762 30$:              CASE    CLI$B_RQTYPE(R9),-              ; DECODE REQUEST TYPE
                            0444   763                         TYPE=B,-                        ; CASE ON A BYTE
                            0444   764                         LIMIT=#CLI$K_NUMERVAL,<-        ; LOWEST REQUEST FOR VALUE LEGAL
                            0444   765                         50$,-                           ; NUMERIC CONVERSION
                            0444   766                         100$,-                          ; ASCII VALUE
                            0444   767                         >
                            044D   768                 SETSTAT INVREQTYP                       ; INCORRECT VALUE
                        04  0452   769 40$:             RET                                     ; EXIT CONVERSION
                51  01      D0  0453   770 50$:             MOVL    #PRC_K_DEC,R1               ; SET RADIX=DECIMAL
                    FBA7'    30  0456   771                 BSBW    DCL$CNVNOEDIT               ; CONVERT NUMERIC
                      06     12  0459   772                 BNEQ    70$                         ; IF NOT EQUAL - CONVERSION ERROR
            0C A9     51     D0  045B   773                 MOVL    R1,CLI$L_RQVALU(R9)         ; SET RESULTANT VALUE
                      1D     11  045F   774                 BRB     120$                        ; EXIT CONVERSION
                             0461   775 70$:              SETSTAT VALCNVERR                       ; SET ERROR
                      6E     B4  0466   776                 CLRW    (SP)                        ; ZERO ANY BYTE COUNT HERE IF ANY
                50    6E     C8  0468   777                 BISL    (SP),R0                     ; INCLUDE INTERNAL BIT IF THERE
                        04   046B   778                 RET                                     ; RETURN TO DISPATCHER
                             046C   779
                             046C   780 ;
                             046C   781 ; REQUEST ASCII STRING VALUE
                             046C   782 ;
                             046C   783
                          3D 3A  046C   784 90$:             .ASCII  \:=\                        ; TERMINATORS FOR KEYVALUES
            08 A9     52     7D  046E   785 100$:            MOVQ    R2,CLI$Q_RQDESC(R9)         ; SET ADDRESS AND SIZE OUTPUT VALUE
        F4 AF   02    6243   3A  0472   786                 LOCC    (R2)[R3],#2,90$             ; CHECK FOR KEY VALUE
                        04   13  0478   787                 BEQL    120$                        ; BR IF NONE LEFT IN MATCH
            03 A9       02   88  047A   788                 BISB    #CLI$M_KEYVALU,CLI$B_RQSTAT(R9) ; INDICATE SUBVALUE FOLLOWING
            04 A7       8E   7D  047E   789 120$:           MOVQ    (SP)+,CLI$Q_QDVALDESC(R7) ; GET DEFAULT VALUE INFORMATION BACK
        03 04 A7       1F   E4  0482   790                 BBSC    #31,CLI$Q_QDVALDESC(R7),140$ ; BR IF DOING DEFAULT VALUE
                    00F6    30  0487   791                 BSBW    DCL$SETQUALVAL              ; SET UP DESCRIPTOR FOR REMAINING VALUE
                52  04 A7     7D  048A   792 140$:           MOVQ    CLI$Q_QDVALDESC(R7),R2      ; GET REMAINING VALUE
                        10   13  048E   793                 BEQL    150$                        ; BR IF THERE IS NONE
            03 A9       01   88  0490   794                 BISB    #CLI$M_MOREVALS,CLI$B_RQSTAT(R9) ; SET FLAG TO INDICATE MORE
                             0494   795                 SETINTR ILLVAL                          ; ASSUME THAT NO MORE ALLOWED
        03 02 A9       00   E0  0498   796                 BBS     #CLI$V_LASTVAL,CLI$B_RQFLGS(R9),RET2 ; BR IF ERROR
                             04A0   797 150$:             SETSTAT SUCCESS                         ; INDICATE GOOD STATUS
                        04   04A3   798 RET2:             RET                                     ; ALL DONE
```

```
                        04A4    800          .SBTTL  PROCESS BIT LISTS
                        04A4    801 ;++
                        04A4    802 ; FUNCTIONAL DESCRIPTION:
                        04A4    803 ;
                        04A4    804 ;       THESE ROUTINES ARE CALLED TO PROCESS THE BIT LISTS SUPPLIED
                        04A4    805 ;       WITH A PARAMETER QUALIFIER. THERE ARE 3 LISTS, THE TEST,SET
                        04A4    806 ;       AND CLEAR LISTS.  THE TEST LIST IS INTENDED TO DETECT CONFLICTING
                        04A4    807 ;       QUALIFIERS AND IS TESTED ONLY WHEN THE QUALIFIER IS FOUND
                        04A4    808 ;       EXPLICITLY TRUE IN THE COMMAND.  THE SET LIST IS SET WHEN THE
                        04A4    809 ;       QUALIFIER IS FOUND TO BE TRUE,CLEARED WHEN THE QUALIFIER IS
                        04A4    810 ;       FOUND TO BE FALSE. THE CLEAR LIST INDICATES A SET OF BITS THAT
                        04A4    811 ;       SHOULD BE CLEARED IF THE QUALFIER IS TRUE.  THIS PERMITS
                        04A4    812 ;       THE PRESENTS OF A QUALIFIER TO OVERRIDE THE PRESENTS OF
                        04A4    813 ;       ANOTHER.
                        04A4    814 ;
                        04A4    815 ; CALLING SEQUENCE:
                        04A4    816 ;
                        04A4    817 ;       BSB/JSB DCL$SETSETLST           ; SET THE SET LIST, CLEAR THE CLEAR LIST
                        04A4    818 ;       BSB/JSB DCL$CLRSETLST           ; CLEAR THE SET LIST, SET THE CLEAR LIST
                        04A4    819 ;       BSB/JSB DCL$TSTSETLST           ; TEST THE TEST LIST, THEN DO SETSETLST
                        04A4    820 ;
                        04A4    821 ; INPUT PARAMETERS:
                        04A4    822 ;
                        04A4    823 ;       R7 CONTAINS THE ADDRESS OF THE PROPER QUALIFIER DESCRIPTOR
                        04A4    824 ;       R8 = ADDRESS OF UTILITY BIT ARRAY
                        04A4    825 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                        04A4    826 ;       R10 = ADDRESS OF WORK BLOCK
                        04A4    827 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                        04A4    828 ;
                        04A4    829 ; OUTPUT PARAMETERS:
                        04A4    830 ;
                        04A4    831 ;       THE BITS ARE SET/CLEARED
                        04A4    832 ;
                        04A4    833 ; SIDE EFFECTS:
                        04A4    834 ;
                        04A4    835 ;       TOP LEVEL ERROR IS ISSUED IF BIT TEST FAILURE
                        04A4    836 ;--
                        04A4    837          .ENABL  LSB
                        04A4    838
                        04A4    839 DCL$TSTSETLST::                         ; TEST THE TEST LIST, THEN DO SET LIST
    52    14 A7   9E    04A4    840          MOVAB   CLI$C_QDBITS(R7),R2    ; GET ADDRESS OF BIT TEST LIST
02 02 A7    01    E1    04A8    841          BBC     #CLI$V_QDUSRV,CLI$B_QDFLGS(R7),5$ ; BR IF NO USER CONTEX VALUE
          82    D5    04AD    842          TSTL    (R2)+                  ; SKIP OVER THE USERS VALUE
       51 82    9A    04AF    843 5$:      MOVZBL  (R2)+,R1               ; GET COUNT OF BITS TO TEST
          0A    13    04B2    844          BEQL    DCL$SETSETLST          ; GO SET THE BITS
       53 82    9A    04B4    845 10$:     MOVZBL  (R2)+,R3               ; GET BIT NUMBER
    36 68 53    E0    04B7    846          BBS     R3,(R8),100$           ; TAKE ERROR EXIT
       F6 51    F5    04BB    847          SOBGTR  R1,10$                 ; BR IF MORE TO DO
                      04BE    848
                      04BE    849 DCL$SETSETLST::                         ; SET THE SET LIST
       50 01    D0    04BE    850          MOVL    #1,R0                  ; SET A TRUE INDICATOR
          06    10    04C1    851          BSBB    50$                    ; PROCESS SET LIST
          50    D4    04C3    852          CLRL    R0                     ; NOW A FALSE
          13    11    04C5    853          BRB     60$                    ; AND DO CLEAR LIST
                      04C7    854
                      04C7    855 DCL$CLRSETLST::                         ; CLEAR THE SET LIST
          50    D4    04C7    856          CLRL    R0                     ; GET A FALSE
```

```
                        04C9    857 ;       BSBB    50$                     ; CLEAR THE SET LIST
                        04C9    858 ;       INCL    R0                      ; NOW TRUE
                        04C9    859 ;       BRB     60$                     ; SET THE CLEAR LIST
                        04C9    860
    52    14 A7   9E    04C9    861 50$:    MOVAB   CLI$C_QDBITS(R7),R2     ; GET ADDRESS OF TEST LIST
02 02 A7    01    E1    04CD    862         BBC     #CLI$V_QDUSRV,CLI$B_QDFLGS(R7),55$ ; BR IF NO USER VALUE PRESENT
          82    D5    04D2    863         TSTL    (R2)+                   ; SKIP USER CONTEX LONGWORD
       51 82    9A    04D4    864 55$:    MOVZBL  (R2)+,R1                ; GET COUNT OF TEST LIST
       52 51    C0    04D7    865         ADDL    R1,R2                   ; AND SKIP OVER THE LIST
       51 82    9A    04DA    866 60$:    MOVZBL  (R2)+,R1                ; GET COUNT OF SET LIST
          11    13    04DD    867         BEQL    90$                     ; BR IF NONE
       53 82    9A    04DF    868 70$:    MOVZBL  (R2)+,R3                ; GET A BIT
                        04E2    869         SETBIT  R3,(R8)                 ; SET THE BIT
         04 50  E8    04E6    870         BLBS    R0,80$                  ; BR IF THAT WAS THE CORRECT ACTION
                        04E9    871         CLRBIT  R3,(R8)                 ; ELSE CLEAR IT
         EF 51  F5    04ED    872 80$:    SOBGTR  R1,70$                  ; DO ALL BITS
                05      04F0    873 90$:    RSB                             ; RETURN
                        04F1    874 ;
                        04F1    875 ; COME HERE WHEN A CONFLICTING QUALIFIER IS FOUND.
                        04F1    876 ; SET ERROR RETURN STRING INFO TO POINT AT THE QUALIFIER
                        04F1    877 ;
                        04F1    878
          55    D4    04F1    879 100$:   CLRL    R5                      ; INIT FOR SEARCH
          55    D6    04F3    880 110$:   INCL    R5                      ; INCREASE INDEX BY 1
     FB08'      30    04F5    881         BSBW    DCL$GETEXTDESC          ; THAT THE DESCRIPTOR APART
                        04F8    882         ASSUME  PTR_K_COMDQUAL EQ 0     ;
                        04F8    883         ASSUME  PTR_K_PARMQUAL EQ 1     ;
01        51    D1    04F8    884         CMPL    R1,#PTR_K_PARMQUAL      ; IS THIS A QUALIFIER
          F6    1A    04FB    885         BGTRU   110$                    ; BR IF NO
    01 A7       91    04FD    886         CMPB    CLI$B_QDCODE(R7),-      ; IS IT THE ONE THAT CONFLICTED?
    05 A4               0500    887                 PTR_B_NUMBER(R4)        ;
          EF    12    0502    888         BNEQ    110$                    ; BR IF NO
                        0504    889         SETSTAT CONFQUAL                ; SET ERROR TO CONFLICTING QUALIFERS
                05      0509    890         RET                             ; REPORT THE ERROR
                        050A    891
                        050A    892         .DSABL  LSB
```

```
                     050A   894          .SBTTL  PROCESS ALL QUALIFIERS IN QUALIFIER LIST
                     050A   895 ;++
                     050A   896 ; FUNCTIONAL DESCRIPTION:
                     050A   897 ;
                     050A   898 ;       THIS ROUTINE IS CALLED WHEN A PARAMETER HAS BEEN FOUND PRESENT
                     050A   899 ;       IN THE COMMAND. THIS ROUTINE SEARCHED FOR ANY COMMAND QUALIFIERS
                     050A   900 ;       PRESENT ITN THE RANGE OF THE COMMAND, WHERE THE RANGE OF THE
                     050A   901 ;       COMMAND IS DEFINED AS ON THE VERB, OR WITHIN THE CURRENT LIMITS
                     050A   902 ;       OF ANY COMMAND PARAMETERS.  ONLY QUALIFIERS EXPLICITLY REQUESTED
                     050A   903 ;       ARE PROCESSED.
                     050A   904 ;
                     050A   905 ; CALLING SEQUENCE:
                     050A   906 ;
                     050A   907 ;       BSB/JSB DCL$,RPOCMDQUAL
                     050A   908 ;
                     050A   909 ; INPUT PARAMETERS:
                     050A   910 ;
                     050A   911 ;       R8 = ADDRESS OF UTILITY BIT ARRAY
                     050A   912 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                     050A   913 ;       R10 = ADDRESS OF WORK BLOCK
                     050A   914 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     050A   915 ;
                     050A   916 ; OUTPUT PARAMETERS:
                     050A   917 ;
                     050A   918 ;       ALL QUALIFIERS SPECIFIED BY THE UTILITY, AND PRESENT ARE PROCESSED.
                     050A   919 ;
                     050A   920 ;--
                     050A   921
                     050A   922 DCL$PROCMDQUAL::                             ; PROCESS COMMAND QUALIFIERS
 0000'CF    9F       050A   923          PUSHAB  W^DCL$FNDCMDQUAL            ; INIT COROUTINE
      9E    16       050E   924 10$:     JSB     @(SP)+                      ; FIND NEXT QUALIFIER IN COMMAND
        46  50       E9 0510   925          BLBC    R0,80$                      ; BR IF NO MORE
   FD2D     CF       9F 0513   926          PUSHAB  W^$CANQUAL                  ; SCAN THE UTILITIES QUALIFIERS
      9E    16       0517   927 20$:     JSB     @(SP)+                      ; FIND NEXT QUALIFIER DESCRIPTOR BLOCK
        F2  50       E9 0519   928          BLBC    R0,10$                      ; BR IF NO MORE UTILITY DESCRIPTORS
        01  A7       91 051C   929          CMPB    CLI$B_QDCODE(R7),-          ; MATCH UTILITY CODE?
        05  A4                 051F   930                  PTR_B_NUMBER(R4)            ;
           F4    12   0521   931          BNEQ    20$                         ; BR IF NO-CKECK UTILITIES NEXT DESCPTR
           8E    D5   0523   932          TSTL    (SP)+                       ; CLR QUAL DESC SCAN COROUTINE
 0070      8F    BB   0525   933          PUSHR   #^M<R4,R5,R6>               ; SAVE INFO USED BY COROUTINE
           10    E0   0529   934          BBS     #CLI$V_ALLOCCUR+<CLI$B_QDFLGS*8>,- ; BR IF UTILITY WANTS TO SEE
        1D  67               052B   935                  (R7),60$                    ; ALL OCCURANCES OF THIS QUALIFIER
                     052D   936          SETBIT  R5,RPW_G_BITS(R10)          ; INDICATE QUALIFIER PROCESSED
        0C  AE       DD 0532   937          PUSHL   12(SP)                      ; COPY COROUTINE ADDRESS
           9E    16   0535   938 30$:     JSB     @(SP)+                      ; CONTINUE SCAN FOR THIS QUALIFIER
        0D  50       E9 0537   939          BLBC    R0,40$                      ; BR IF NO MORE OCCURANCES
        01  A7       91 053A   940          CMPB    CLI$B_QDCODE(R7),-          ; IS THIS THE SAME QUALIFIER?
        05  A4                 053D   941                  PTR_B_NUMBER(R4)            ;
           F4    12   053F   942          BNEQ    30$                         ; IF NO LOOK SOME MORE
 0071      8F    BA   0541   943          POPR    #^M<R0,R4,R5,R6>            ; POP RETURN ADDRESS PLUS PARAMETERS
           C7    11   0545   944          BRB     10$                         ; PROCESS THIS WHEN WE FIND IT AGAIN
        54  6E    7D 0547   945 40$:     MOVQ    (SP),R4                     ; SET THE VALUE OF QUALIFIER DESCRIPTOR
           0E    10   054A   946 60$:     BSBB    DCL$HANDLQUAL               ; HANDLE THE QUALIFIER
 0070      8F    BA   054C   947          POPR    #^M<R4,R5,R6>               ; RESTORE INFO USED BY COROUTINE
BA 67       10    E1 0550   948          BBC     #CLI$V_ALLOCCUR+<CLI$B_RQFLGS*8>,(R7),10$ ; DOING ALL OCCURANCES
     FCB7     30    0554   949          BSBW    QUALACT                     ; IF YES TAKE ACTION AT THIS TIME
           B5    11   0557   950          BRB     10$                         ; LOOK FOR MORE
```

```
05  0559    951 80$:    RSB                                 ;
```

```
                     055A   953          .SBTTL  PROCESS QUALIFIER
                     055A   954 ;++
                     055A   955 ; FUNCTIONAL DESCRIPTION:
                     055A   956 ;
                     055A   957 ;       THIS ROUTINE IS CALLED TO PROCESS A QUALIFIER FOUND IN THE
                     055A   958 ;       COMMAND LINE, AND SET ALL UTILITY STRUCTURES CORRECTLY.
                     055A   959 ;
                     055A   960 ; CALLING SEQUENCE:
                     055A   961 ;
                     055A   962 ;       BSB/JSB DCL$HANDLQUAL
                     055A   963 ;
                     055A   964 ; INPUT PARAMETERS:
                     055A   965 ;
                     055A   966 ;       R4 CONTAINS THE ADDRESS OF THE RESULT PARSE DESCRIPTOR FOR THE QUALIFIER
                     055A   967 ;       R5 IS INDEX TO THE RESULT DESCRIPTOR FOR THE QUALFIER
                     055A   968 ;       R7 CONTAINS THE ADDRESS OF THE UTILITY QUALIFIER DESCRIPTOR
                     055A   969 ;
                     055A   970 ; IMPLICIT INPUTS:
                     055A   971 ;
                     055A   972 ;       R8 = ADDRESS OF UTILITY BIT ARRAY
                     055A   973 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                     055A   974 ;       R10 = ADDRESS OF WORK BLOCK
                     055A   975 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     055A   976 ;
                     055A   977 ; OUTPUT PARAMETERS:
                     055A   978 ;
                     055A   979 ;       UTILITY QUALIFER DATA STRUTURE IS SET PROPERLY
                     055A   980 ;
                     055A   981 ;--
                     055A   982          .ENABL  LSB
                     055A   983
                     055A   984 DCL$HANDLQUAL::                             ; PROCESS A QUALIFIER
          04 A7   7C 055A   985          CLRQ    CLI$Q_QDVALDESC(R7)       ; SET VALUE TO NONE
                     055D   986          SETBIT  R5,RPG_G_BITS(R10)        ; COUNT THIS QUALIFIER AS PROCESSED
             02   88 0562   987          BISB    #CLI$M_QUALEXP,-          ; SET FLAG TO INDICATE QUALIFIER WAS
          03 A7       0564   988                  CLI$B_QDSTAT(R7)          ; EXPLICITLY FOUND
       03 A7   01   8A 0566   989          BICB    #CLI$M_QUALTRU,CLI$B_QDSTAT(R7) ; AND SET STATE TO FALSE
             14   E0 056A   990          BBS     #PTR_V_NEGATE,-           ; BR IF THE ASSUMED STATE,FALSE,
          3C 64       056C   991                  PTR_L_DESCR(R4),40$       ; BR IF ASSUMED CORRECTLY
             01   88 056E   992          BISB    #CLI$M_QUALTRU,-          ; ASSUMED INCORRECTLY, SET STATE OF
          03 A7       0570   993                  CLI$B_QDSTAT(R7)          ; QUALIFIER TO TRUE
          04   18   ED 0572   994          CMPZV   #PTR_V_TERM,#PTR_S_TERM,- ; TERMINATOR VIELD LIMITS
          02   64       0575   995                  PTR_L_DESCR(R4),#PTR_K_COLON ; EXPLICIT VALUE GIVEN?
                07   13 0577   996          BEQL    DCL$SETQUALVAL            ; BR IF YES, SET USER SPECIFIED VALUE
          04   18   ED 0579   997          CMPZV   #PTR_V_TERM,#PTR_S_TERM,- ; TERMINATOR VIELD LIMITS
          07   64       057C   998                  PTR_L_DESCR(R4),#PTR_K_LPAREN ; EXPLICIT VALUE GIVEN?
                2B   12 057E   999          BNEQ    70$                       ; BR IF NO, SET DEFAULT IF THERE IS ONE
                     0580  1000 ; DROP THRU TO RETURN EXPLICIT OR DEFAULT VALUE (IF ANY)
```

```
                        0580  1002          .SBTTL  RETURN EXPLICIT QUALIFIER VALUE
                        0580  1003 ;++
                        0580  1004 ; FUNCTIONAL DESCRIPTION:
                        0580  1005 ;
                        0580  1006 ;       THIS ROUTINE IS CALLED TO SET THE STRING LIMITS OF
                        0580  1007 ;       A EXPLICIT VALUE ENTERED VIA THE COMMAND STREAM.
                        0580  1008 ;
                        0580  1009 ; CALLING SEQUENCE:
                        0580  1010 ;
                        0580  1011 ;       BSB/JSB DCL$SETQUALVAL
                        0580  1012 ;
                        0580  1013 ; INPUT PARAMETERS:
                        0580  1014 ;
                        0580  1015 ;       R5 IS INDEX TO THE RESULT DESCRIPTOR FOR THE QUALFIER OR LAST VALUE
                        0580  1016 ;       R7 CONTAINS THE ADDRESS OF THE UTILITY QUALIFIER DESCRIPTOR
                        0580  1017 ;
                        0580  1018 ; IMPLICIT INPUTS:
                        0580  1019 ;
                        0580  1020 ;       R8 = ADDRESS OF UTILITY BIT ARRAY
                        0580  1021 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                        0580  1022 ;       R10 = ADDRESS OF WORK BLOCK
                        0580  1023 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                        0580  1024 ;
                        0580  1025 ; OUTPUT PARAMETERS:
                        0580  1026 ;
                        0580  1027 ;       UTILITY QUALIFER DATA STRUTURE IS SET PROPERLY
                        0580  1028 ;
                        0580  1029 ;---
                        0580  1030 DCL$SETQUALVAL:                          ; SET QUALIFIER VALUE ONLY
      04 A7    7C       0580  1031          CLRQ    CLI$Q_QDVALDESC(R7)     ; ASSUME NO VALUE PRESENT
         55    D6       0583  1032          INCL    R5                      ; ADV INDEX TO NEXT RESULT DESCRIPTOR
       FA78'   30       0585  1033          BSBW    DCL$GETEXTDESC          ; TAKE THAT 1 APART
   02    51    91       0588  1034          CMPB    R1,#PTR_K_QUALVALU      ; WAS THIS A VALUE?
         1D    12       058B  1035          BNEQ    40$                     ; BR IF NO
         0C    BB       058D  1036          PUSHR   #^M<R2,R3>              ; SET CURRENT LIMIT VALUES
         0A    11       058F  1037          BRB     20$                     ; JOIN COMMON LOOP
         55    D6       0591  1038 10$:     INCL    R5                      ; ADD 1 TO INDEX INTO RESULT BUFFER
       FA6A'   30       0593  1039          BSBW    DCL$GETEXTDESC          ; TAKE THE DESCRIPTOR APART
   02    51    91       0596  1040          CMPB    R1,#PTR_K_QUALVALU      ; LAST VALUE IN LIST?
         06    12       0599  1041          BNEQ    30$                     ; BR IF YES-EXIT THE LOOP
6E    53    52    C1   059B  1042 20$:     ADDL3   R2,R3,(SP)              ; FIND END OF LAST VALUE
         F0    11       059F  1043          BRB     10$                     ; LOOK FOR MORE
         0C    BA       05A1  1044 30$:     POPR    #^M<R2,R3>              ; GET VALUE LIMITS BACK
   52    53    C2       05A3  1045          SUBL    R3,R2                   ; CHANGE TO LENGTH
04 A7    52    7D       05A6  1046          MOVQ    R2,CLI$Q_QDVALDESC(R7)  ; SET VALUE
               05       05AA  1047 40$:     RSB                             ; PROCESS BIT LISTS-RETURN FROM THERE
                        05AB  1048
   03    69    91       05AB  1049 70$:     CMPB    (R9),#CLI$K_GETOPT      ; IS THIS AN OPTIONS PARSE
         1B    13       05AE  1050          BEQL    80$                     ; BR IF SO - NO DEFAULT VALUES THEN
51    05 A4    9A       05B0  1051          MOVZBL  PTR_B_NUMBER(R4),R1     ; GET QUALIFIER NUMBER
       FA49'   30       05B4  1052          BSBW    DCL$GETPARMQUAL         ; LOCATE ASSOCIATED QUALIFER BLOCK
                        05B7  1053 ; DROP THRU TO RETURN THE QUALIFIER DEFAULT VALUE (IF ANY)
```

```
                     05B7  1055          .SBTTL  RETURN QUALIFIER DEFAULT VALUE
                     05B7  1056 ;++
                     05B7  1057 ; FUNCTIONAL DESCRIPTION:
                     05B7  1058 ;
                     05B7  1059 ;       THIS ROUTINE IS CALLED TO SET THE STRING LIMITS FOR
                     05B7  1060 ;       A DEFAULT VALUE ASSOCIATED WITH A QUALIFER THAT IS TRUE.
                     05B7  1061 ;
                     05B7  1062 ; CALLING SEQUENCE:
                     05B7  1063 ;
                     05B7  1064 ;       BSB/JSB DCL$SETDEFVAL
                     05B7  1065 ;
                     05B7  1066 ; INPUT PARAMETERS:
                     05B7  1067 ;
                     05B7  1068 ;       R2 CONTAINS THE ADDRESS OF DCL INTERNAL QUALIFER DESCRIPTOR
                     05B7  1069 ;       R7 CONTAINS THE ADDRESS OF THE UTILITY QUALIFIER DESCRIPTOR
                     05B7  1070 ;
                     05B7  1071 ; IMPLICIT INPUTS:
                     05B7  1072 ;
                     05B7  1073 ;       R8 = ADDRESS OF UTILITY BIT ARRAY
                     05B7  1074 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                     05B7  1075 ;       R10 = ADDRESS OF WORK BLOCK
                     05B7  1076 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     05B7  1077 ;
                     05B7  1078 ; OUTPUT PARAMETERS:
                     05B7  1079 ;
                     05B7  1080 ;       UTILITY QUALIFER DATA STRUTURE IS SET PROPERLY
                     05B7  1081 ;
                     05B7  1082 ;---
                     05B7  1083 DCL$SETDEFVAL:                          ; RETURN QUALIFER DEFAULT VALUE
50   1C A2      32   05B7  1084          CVTWL   ENT_W_DEFVAL(R2),R0     ; GET OFFSET TO DEFAULT VALUE STRING
           0E   13   05BB  1085          BEQL    80$                     ; BR IF NO DEFAULT VALUE
     50    01   C0   05BD  1086          ADDL    #1,R0                   ; FIND REAL ADDRESS OF DEFAULT VALUE
     50    52   C0   05C0  1087          ADDL    R2,R0                   ;
04 A7      80   9B   05C3  1088          MOVZBW  (R0)+,CLI$W_QDVALSIZ(R7) ; SET SIZE OF VALUE STRING
08 A7      50   D0   05C7  1089          MOVL    R0,CLI$A_QDVALADR(R7)   ; AND THE ADDRESS OF THE STRING
                05   05CB  1090 80$:     RSB                             ; RETURN FROM DEFAULT VALUE PROCESSING
                     05CC  1091
                     05CC  1092          .DSABL  LSB
```

```
                        05CC  1094          .SBTTL  GET OPTION VALUE
                        05CC  1095 ;++
                        05CC  1096 ; FUNCTIONAL DESCRIPTION:
                        05CC  1097 ;
                        05CC  1098 ;       AN OPTION IS A DCL COMMAND PARAMETER/QUALIFIER.  IT MUST
                        05CC  1099 ;       BE THE FIRST ENTITY FOLLOWING THE VERB.  THIS ROUTINE IS
                        05CC  1100 ;       CALLED BY AN IMAGE THAT HAS SEVERAL OPTIONS TO PROCESS AND
                        05CC  1101 ;       WOULD LIKE TO BE TOLD WHICH IT IS TO DO.  OPTIONS APPEAR IN
                        05CC  1102 ;       THE RESULT PARSE BUFFER AS THE FIRST ENTRY AND AS
                        05CC  1103 ;       PARAMETERS.  THE ONLY OUTPUT OF THIS ROUTINE
                        05CC  1104 ;       IS THE EXECUTION OF THE ACTION ROUTINE FOR THE OPTION.
                        05CC  1105 ;       FAILURE TO SPECIFY ACTION ROUTINES FOR OPTIONS RESULTS
                        05CC  1106 ;       IN CAUSING THIS CALL BACK TO BE A NO-OP.
                        05CC  1107 ;
                        05CC  1108 ; CALLING SEQUENCE:
                        05CC  1109 ;
                        05CC  1110 ;       ENTERED VIA A CASE FOLLOWING A CALL
                        05CC  1111 ;
                        05CC  1112 ; INPUT PARAMETERS:
                        05CC  1113 ;
                        05CC  1114 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR FOR VALUE CONVERSION
                        05CC  1115 ;       R10 = ADDRESS OF IMAGE LOCAL WORK AREA
                        05CC  1116 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                        05CC  1117 ;
                        05CC  1118 ; OUTPUT PARAMETERS:
                        05CC  1119 ;
                        05CC  1120 ;       THE OPTION QUALIFER ACTION ROUTINE IS EXECUTED FOR THE QUALIFIER
                        05CC  1121 ;       THAT MATCHES THE CODE.
                        05CC  1122 ;
                        05CC  1123 ; COMPLETION CODES:
                        05CC  1124 ;
                        05CC  1125 ;       DCL$INVQUAL IF NO MATCH ON THE QUALIFIER CODE
                        05CC  1126 ;           ELSE AS SET BY THE OPTION ACTION ROUTINE.
                        05CC  1127 ;
                        05CC  1128 ;--
                        05CC  1129 DCL$GETOPT:                                  ; FIND COMMAND OPTION
                        05CC  1130          SETSTAT <NOOPTPRS>                  ; ASSUME NO OPTION PRESENT
          C3 AB    95   05D1  1131          TSTB    WRK_B_CMDOPT(R11)           ; TEST KEYWORD/QUALIFIER NUMBER CAUSING CHAN
             32    13   05D4  1132          BEQL    20$                         ; IF ZERO-THIS COMMAND HAS NO OPTIONS
54   F9B6 CB    9E   05D6  1133          MOVAB   WRK_G_RESULT(R11),R4        ; SET ADDRESS OF FIRST TOKEN DESCRIPTOR
             1C    ED   05DB  1134 2$:      CMPZV   #PTR_V_TYPE,-               ; END OF RESULT DESCRIPTOR ARRAY?
             04         05DD  1135                  #PTR_S_TYPE,-               ;
     04      64         05DE  1136                  (R4),#PTR_K_ENDLINE         ;
             26    13   05E0  1137          BEQL    20$                         ; YES, THEN EXIT
  07 64      16    E1   05E2  1138          BBC     #PTR_V_SYNTAX,(R4),6$       ; BRANCH IF NOT TOKEN CAUSING A CHANGE
        05   A4    91   05E6  1139          CMPB    PTR_B_NUMBER(R4),-          ; IS IT THE ONE WE WANT?
          C3 AB         05E9  1140                  WRK_B_CMDOPT(R11)           ;
             05    13   05EB  1141          BEQL    8$                          ; YES, THEN EXIT LOOP
     54      0C    C0   05ED  1142 6$:      ADDL    #PTR_C_LENGTH,R4            ; GET NEXT DESCRIPTOR
             E9    11   05F0  1143          BRB     2$                          ; AND LOOP
     FC4E    CF    9F   05F2  1144 8$:      PUSHAB  SCANQUAL                    ; SET COROUTINE TO SCAN INPUT QUALIFERS
             9E    16   05F6  1145 10$:     JSB     @(SP)+                      ; GET CALLERS NEXT QUALIFIER DESCRIPTOR
        0D   50    E9   05F8  1146          BLBC    R0,20$                      ; BR IF NOT FOUND
          C3 AB    91   05FB  1147          CMPB    WRK_B_CMDOPT(R11),-         ; IS THIS THE QUALIFIER HE WANTED?
        01   A7         05FE  1148                  CLI$B_QDCODE(R7)            ;
             F4    12   0600  1149          BNEQ    10$                         ; BR IF NO-KEEP LOOKING
           FF55    30   0602  1150          BSBW    DCL$HANDLQUAL               ; SET USERS STRUCTURE
```

```
FC06    30  0605  1151          BSBW    QUALACT                 ; TAKE PROPER ACTION
        04  0608  1152 20$:      RET                             ; RETURN TO DISPATCHER
```

```
                     0609  1154          .SBTTL  GET COMMAND LINE
                     0609  1155 ;++
                     0609  1156 ; FUNCTIONAL DESCRIPTION:
                     0609  1157 ;
                     0609  1158 ;       THIS ROUTINE IS CALLED TO SET A DESCRITOR FOR THE COMMAND
                     0609  1159 ;       THAT WAS JUST PROCESSED BY DCL.
                     0609  1160 ;
                     0609  1161 ; CALLING SEQUENCE:
                     0609  1162 ;
                     0609  1163 ;       THIS ROUTINE IS ENTERED BY A CASE FOLLOWING A CALL
                     0609  1164 ;
                     0609  1165 ; INPUT PARAMETERS:
                     0609  1166 ;
                     0609  1167 ;       R9 = ADDRESS OF REQUEST DESCRIPTOR
                     0609  1168 ;       R11 = ADDRESS OF PASS 1 PARSE WORK AREA
                     0609  1169 ;
                     0609  1170 ; OUTPUT PARAMETERS:
                     0609  1171 ;
                     0609  1172 ;       THE REQUEST DESCRIPTOR IS SET TO CONTAIN A QUADWORD DESCRIPTOR
                     0609  1173 ;       THE THE FINAL COMMAND IN THE BUFFER.
                     0609  1174 ;
                     0609  1175 ; IMPLICIT OUTPUTS:
                     0609  1176 ;
                     0609  1177 ;       THE INTERNAL ERROR MECHANISM IS USED TO RETURN THE RESULTANT
                     0609  1178 ;       COMMAND LINE DESCRIPTOR WHEN COMMAND IS A RUN
                     0609  1179 ;
                     0609  1180 ; COMPLETION CODES:
                     0609  1181 ;
                     0609  1182 ;       SUCCESS IN ALL CASES EXCEPT WHEN COMMAND IS A "RUN".  IN THIS
                     0609  1183 ;       WAY, A UTILITY MAY DETERMIN THAT IS WAS INVOKED VIA A COMMAND,
                     0609  1184 ;       IE: LINK ALPHA, OR BY A "RUN FILESPEC".
                     0609  1185 ;
                     0609  1186 ;--
                     0609  1187 DCL$GETCMD::                                ; GET COMMAND LINE
          F9F4'  30  0609  1188          BSBW    DCL$GETDCLWRK             ; SET WORK AREA POINTER
03 A9     C2 AB   90  060C  1189          MOVB    WRK_B_VERBTYP(R11),CLI$B_RQSTAT(R9); GET VERB TYPE FOR CALLER
             55   D4  0611  1190          CLRL    R5                        ; START AT FIRST TOKEN
          F9EA'  30  0613  1191          BSBW    DCL$SETDESCADR            ; SET ADDRESS OF TOKEN DESCRIPTOR
    52    B6 AB   D0  0616  1192          MOVL    WRK_L_RSLEND(R11),R2      ; GET ADDRESS OF NEXT FREE DESCRIPTOR
 0C       08   EF  061A  1193          EXTZV   #PTR_V_OFFSET,#PTR_S_OFFSET,- ; GET OFFSET TO EOL
    52    F4 A2       061D  1194                  -PTR_C_LENGTH(R2),R2
       53    52   D0  0620  1195          MOVL    R2,R3                     ; PRESET FIRST TOKEN TO EOL
       54    0C   C0  0623  1196 10$:     ADDL    #PTR_C_LENGTH,R4          ; SKIP TO NEXT TOKEN
04   64    04   1C   ED  0626  1197          CMPZV   #PTR_V_TYPE,#PTR_S_TYPE,(R4),#PTR_K_ENDLINE ; END OF LINE?
             0E   13  062B  1198          BEQL    20$                       ; BRANCH IF DONE
53   64    0C   08   ED  062D  1199          CMPZV   #PTR_V_OFFSET,#PTR_S_OFFSET,(R4),R3 ; FIRST TOKEN IN COMMAND?
             EF   1E  0632  1200          BGEQU   10$                       ; BRANCH IF NOT
53   64    0C   08   EF  0634  1201          EXTZV   #PTR_V_OFFSET,#PTR_S_OFFSET,(R4),R3 ; SET OFFSET TO FIRST TOKEN
             E8   11  0639  1202          BRB     10$                       ; PRECEEDS FIRST, SET IT AS NEW FIRST
       52    53   C2  063B  1203 20$:     SUBL    R3,R2                     ; FIND LENGTH OF COMMAND
   53   F492 CB43   9E  063E  1204          MOVAB   WRK_G_BUFFER(R11)[R3],R3 ; GET ADDRESS OF FIRST TOKEN
     FF A3    2F   91  0644  1205          CMPB    #^A\/\,-1(R3)             ; COMMAND TERMINATOR A SLASH?
             04   12  0648  1206          BNEQ    30$                       ; IF NOT-THEN DON'T INCLUE IT
             52   D6  064A  1207          INCL    R2                        ; ADD 1 TO COUNT
             53   D7  064C  1208          DECL    R3                        ; BACK UP ADDRESS TO TERMINATOR
    08 A9    52   7D  064E  1209 30$:     MOVQ    R2,CLI$Q_RQDESC(R9)       ; SET RESULT IN CALLER DATA BLOCK
50   80000000 8F   D0  0652  1210          MOVL    #1@31,R0                  ; SET INTERNAL ERROR BIT
```

```
                03 A9   24   91  0659  1211          CMPB    #CLI$K_VERB_RUN,CLI$B_RQSTAT(R9) ; WAS COMMAND A RUN?
                       07   13  065D  1212          BEQL    90$                              ; IF YES - THERE IS NO COMMAND LINE
                                  065F  1213          STATUS  NORMAL                           ; SET GOOD STATUS
                            04    0666  1214 90$:     RET                                      ; RETURN TO DISPATCHER
                                  0667  1215
                                  0667  1216          .END
```

```
CALERR                          00000040 R      02
CALLBAK                         0000022A R      02
CLI$A_ABSACT                =   00000014
CLI$A_ERRACT                =   00000004
CLI$A_FLSACT                =   00000010
CLI$A_PRSACT                =   00000010
CLI$A_QDVALADR              =   00000008
CLI$A_QUALST                =   00000018
CLI$A_TRUACT                =   0000000C
CLI$B_BITNUM                =   00000001
CLI$B_QDBLKSIZ              =   00000000
CLI$B_QDCODE                =   00000001
CLI$B_QDFLGS                =   00000002
CLI$B_QDSTAT                =   00000003
CLI$B_RQFLGS                =   00000002
CLI$B_RQSTAT                =   00000003
CLI$B_RQTYPE                =   00000000
CLI$C_QDBITS                =   00000014
CLI$GET_PRC                     ********    X   02
CLI$K_CLINT                 =   00000005
CLI$K_CLISERV               =   00000005
CLI$K_GETOPT                =   00000003
CLI$K_GETQUAL               =   00000002
CLI$K_INITPRS               =   00000000
CLI$K_INPSPEC               =   00000001
CLI$K_NUMERVAL              =   00000040
CLI$K_OUTSPEC               =   00000002
CLI$K_PRESENT               =   00000050
CLI$K_VERB_RUN              =   00000024
CLI$L_RQVALU                =   0000000C
CLI$M_CONCATINP             =   00000002
CLI$M_KEYVALU               =   00000002
CLI$M_MOREINP               =   00000004
CLI$M_MOREVALS              =   00000001
CLI$M_PARMDEF               =   00000008
CLI$M_PARMPRS               =   00000001
CLI$M_QUALEXP               =   00000002
CLI$M_QUALTRU               =   00000001
CLI$Q_QDVALDESC             =   00000004
CLI$Q_RQDESC                =   00000008
CLI$S_PRITYP                =   00000004
CLI$S_SUBTYP                =   00000004
CLI$V_ABSADR                =   00000001
CLI$V_ALLOCCUR              =   00000000
CLI$V_EXPNAM                =   00000002
CLI$V_LASTVAL               =   00000000
CLI$V_PARMPRS               =   00000000
CLI$V_PARMREQ               =   00000000
CLI$V_PRITYP                =   00000004
CLI$V_QDEXPA                =   00000002
CLI$V_QDUSRV                =   00000001
CLI$V_QUALEXP               =   00000001
CLI$V_QUALTRU               =   00000000
CLI$V_SUBTYP                =   00000000
CLI$W_QDVALSIZ              =   00000004
CLI$W_RQSIZE                =   00000008
CLI$W_SERVCOD               =   00000001
CLI$_ABKEYW                 =   00038010
CLI$_CONFQUAL               =   00038802
CLI$_ILLVAL                 =   0003883A
CLI$_INVQUAL                =   0003880A
CLI$_INVREQTYP              =   00038822
CLI$_NOOPTPRS               =   00038842
CLI$_NORMAL                 =   00030001
CLI$_NOVALUE                =   0003882A
CLI$_REQPRMABS              =   00038812
CLI$_UNPROPARM              =   00038170
CLI$_UNPROQUAL              =   00038168
CLI$_VALCNVERR              =   00038832
CLINT                           000000BA R      02
CMD_K_MAX_PARMS             =   00000008
CMPPRM                          000000CC R      02
DCL$CLRSETLST                   000004C7 RG     02
DCL$CNVNOEDIT                   ********   X    02
DCL$DCLPARSE                    ********   X    02
DCL$DISPATCH                    ********   X    02
DCL$ENDPARSE                    ********   X    02
DCL$EXTNXTDESC                  ********   X    02
DCL$EXTRSLDESC                  ********   X    02
DCL$FNDCMDQUAL                  ********   X    02
DCL$GETCMD                      00000609 RG     02
DCL$GETDCLWRK                   ********   X    02
DCL$GETEXTDESC                  ********   X    02
DCL$GETLINE                     ********   X    02
DCL$GETOPT                      000005CC R      02
DCL$GETPARM                     ********   X    02
DCL$GETPARMQUAL                 ********   X    02
DCL$GETQUALDESC                 ********   X    02
DCL$GETVALUE                    ********   X    02
DCL$HANDLQUAL                   0000055A RG     02
DCL$NEXTQUAL                    ********   X    02
DCL$PRESENT                     ********   X    02
DCL$PROCMDQUAL                  0000050A RG     02
DCL$RPINIT                      ********   X    02
DCL$SETDEFVAL                   000005B7 R      02
DCL$SETDESCADR                  ********   X    02
DCL$SETQUALVAL                  00000580 R      02
DCL$SETSETLST                   000004BE RG     02
DCL$TSTSETLST                   000004A4 RG     02
DCL$UTLSERV                     00000000 RG     02
DCL$VALCNV                      000003E8 R      02
ENT_V_BATDEF                =   00000003
ENT_V_DEFTRUE               =   00000002
ENT_W_DEFVAL                =   0000001C
ENT_W_FLAGS                 =   00000004
INPUT                           00000398 R      02
INTERROR                    =   0000001F
OUTPUT                          00000277 R      02
PLM_B_FSTDESC                   00000001
PLM_B_LSTDESC                   00000002
PLM_B_NXTDESC                   00000000
PLM_B_QUADESC                   00000003
PLM_B_TRMDESC                   00000003
PLM_C_SIZE                      00000004
```

```
PLM_K_SIZE                      00000004
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_DEC                     = 00000001
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                       000000E8
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_V_MODE                    = 00000006
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
PROCIO                          00000268 R      02
PTR_B_LEVEL                     00000004
PTR_B_NUMBER                    00000005
PTR_B_PARMCNT                   00000006
PTR_B_VALUE                     00000000
PTR_C_LENGTH                    0000000C
PTR_K_BLANK                   = 00000001
PTR_K_COLON                   = 00000002
PTR_K_COMDQUAL                = 00000000
PTR_K_COMMA                   = 00000005
PTR_K_ENDLINE                 = 00000004
PTR_K_LENGTH                    0000000C
PTR_K_LPAREN                  = 00000007
PTR_K_PARAMETR                = 00000003
PTR_K_PARMQUAL                = 00000001
PTR_K_QUALVALU                = 00000002
PTR_L_DESCR                     00000000
PTR_L_ENTITY                    00000008
PTR_S_OFFSET                  = 0000000C
PTR_S_TERM                    = 00000004
PTR_S_TYPE                    = 00000004
PTR_V_NEGATE                  = 00000014
PTR_V_OFFSET                  = 00000008
PTR_V_SYNTAX                  = 00000016
PTR_V_TERM                    = 00000018
PTR_V_TYPE                    = 0000001C
QUA[ACT                         0000020E R      02
RET0                            00000089 R      02
RET1                            000003E7 R      02
RET2                            000004A3 R      02
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
RQBITS                        = 0000000C
RQDESC                        = 00000004
RQWORK                        = 00000008
RSBO                            00000267 R      02
RSLTPRS                         0000008A R      02
SCANQUAL                        00000244 R      02
SETQUAL                         0000015A 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
_$$_                          = 00000055
```

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	00000667 (1639.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	16	00:00:00.07	00:00:00.77
Command processing	97	00:00:00.63	00:00:07.33
Pass 1	310	00:00:12.67	00:00:43.11
Symbol table sort	0	00:00:01.56	00:00:04.85
Pass 2	217	00:00:03.33	00:00:12.84
Symbol table output	31	00:00:00.24	00:00:01.19
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	673	00:00:18.54	00:01:10.12

The working set limit was 1650 pages.
68000 bytes (133 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 914 non-local and 106 local symbols.
1216 source lines were read in Pass 1, producing 21 object records in Pass 2.
43 pages of virtual memory were used to define 29 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
_255DUA28:[DCL.OBJ]DCL.MLB;1	13
_255DUA28:[SYS.OBJ]LIB.MLB;1	2
_255DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	21

1079 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

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

SHOW
LIS

RPDCL
LIS

SET
LIS

SPAWN
LIS

RPSUB
LIS