```
DDDDDDDDDDD          CCCCCCCCCCC  LLL
DDDDDDDDDDD          CCCCCCCCCCC  LLL
DDDDDDDDDDD          CCCCCCCCCCC  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
DDDDDDDDDDD          CCCCCCCCCCC  LLLLLLLLLLLLLL
DDDDDDDDDDD          CCCCCCCCCCC  LLLLLLLLLLLLLL
DDDDDDDDDDD          CCCCCCCCCCC  LLLLLLLLLLLLLL
```

```
  CCCCCCCC    000000   NN       NN  NN       NN  EEEEEEEEEE   CCCCCCCC  TTTTTTTTTT
  CCCCCCCC    000000   NN       NN  NN       NN  EEEEEEEEEE   CCCCCCCC  TTTTTTTTTT
  CC         00    00  NN       NN  NN       NN  EE           CC            TT
  CC         00    00  NN       NN  NN       NN  EE           CC            TT
  CC         00    00  NNNN     NN  NNNN     NN  EE           CC            TT
  CC         00    00  NNNN     NN  NNNN     NN  EE           CC            TT
  CC         00    00  NN  NN   NN  NN  NN   NN  EEEEEEEE     CC            TT
  CC         00    00  NN  NN   NN  NN  NN   NN  EEEEEEEE     CC            TT
  CC         00    00  NN    NNNN   NN    NNNN   EE           CC            TT
  CC         00    00  NN    NNNN   NN    NNNN   EE           CC            TT
  CC         00    00  NN       NN  NN       NN  EE           CC            TT        ....
  CC         00    00  NN       NN  NN       NN  EE           CC            TT        ....
  CCCCCCCC    000000   NN       NN  NN       NN  EEEEEEEEEE   CCCCCCCC      TT        ....
  CCCCCCCC    000000   NN       NN  NN       NN  EEEEEEEEEE   CCCCCCCC      TT        ....


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

```
0000     1            .TITLE  CONNECT - Connect/disconnect process to terminal
0000     2            .IDENT  'V04-000'
0000     3
0000     4    ;********************************************************************
0000     5    ;*                                                                  *
0000     6    ;*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                        *
0000     7    ;*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.         *
0000     8    ;*   ALL RIGHTS RESERVED.                                           *
0000     9    ;*                                                                  *
0000    10    ;*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000    11    ;*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000    12    ;*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000    13    ;*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000    14    ;*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000    15    ;*   TRANSFERRED.                                                   *
0000    16    ;*                                                                  *
0000    17    ;*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000    18    ;*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000    19    ;*   CORPORATION.                                                   *
0000    20    ;*                                                                  *
0000    21    ;*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000    22    ;*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.        *
0000    23    ;*                                                                  *
0000    24    ;*                                                                  *
0000    25    ;********************************************************************
0000    26    ;
00C0    27    ; ABSTRACT:
0000    28    ;
0000    29    ;        Connect/disconnect process to terminal
0000    30    ;
0000    31    ; AUTHOR:
0000    32    ;
0000    33    ;        Peter George,  June 1983
0000    34    ;
0000    35    ; MODIFIED BY:
0000    36    ;
0000    37    ;        V03-001 PCG0001        Peter George            27-Jun-1983
0000    38    ;                Use CONT as name of ASCIC string, not CONTINUE.
0000    39    ;                Use event flags more intelligently.
0000    40    ;---
0000    41
0000    42    ;
0000    43    ; Macro library calls
0000    44    ;
0000    45            PRCDEF                          ; Define process data structure
0000    46            WRKDEF                          ; Define command data structure
0000    47            PTRDEF                          ; Define token descriptor fields
0000    48            $CLIMSGDEF                      ; Define cli related errors
0000    49            $DEVDEF                         ; Define device characteristics
0000    50            $DVIDEF                         ; Define $GETDVI codes
0000    51            $IODEF                          ; Define $QIO codes
0000    52            $RABDEF                         ; Define RAB fields
0000    53            $TT2DEF                         ; Define terminal characteristics
0000    54
00000000 55            .PSECT  DCL$ZCODE,BYTE,RD,NOWRT
0000    56
0000    57 CONT:
```

```
                             09  0000   58               .BYTE   9
      45 55 4E 49 54 4E 4F  43  0001   59               .ASCII  /CONTINUE/
                             00  0009   60               .BYTE   0
```

CONNECT
V04-000

J 9
- Connect/disconnect process to terminal 15-SEP-1984 23:41:17 VAX/VMS Macro V04-00   Page 3
Connect process to terminal   4-SEP-1984 23:39:49 [DCL.SRC]CONNECT.MAR;1   (2)

```
                              000A        62              .SBTTL   Connect process to terminal
                              000A        63       ;+
                              000A        64       ; DCL$CONNECT - Connect process to terminal
                              000A        65       ;
                              000A        66       ; This routine is called as an internal command to execute the CONNECT
                              000A        67       ; DCL command.
                              000A        68       ;
                              000A        69       ; INPUTS:
                              000A        70       ;
                              000A        71       ;      R8 = Address of scratch buffer descriptor.
                              000A        72       ;      R9 = Address of scratch stack.
                              000A        73       ;      R10 = Base address of command work area.
                              000A        74       ;      R11 = Base address of process work area.
                              000A        75       ;
                              000A        76       ; OUTPUTS:
                              000A        77       ;
                              000A        78       ;      The process is connected to another terminal.
                              000A        79       ;-
                              000A        80       ;
                              000A        81       DCL$CONNECT::                                          ; Connect process
                              000A        82       ;
                              000A        83       ; Exit if batch job or input is not from a terminal.
                              000A        84       ;
                              000A        85              STATUS   NOTDISCON                     ; Assume not disconnectable
        79 68 AB   06   E0    0011        86              BBS      #PRC_V_MODE,PRC_W_FLAGS(R11),80$; Exit if batch
           51   08 AB   D0    0016        87              MOVL     PRC_L_INPRAB(R11),R1          ; Get addr of input RAB
        70 18 A1   02   E1    001A        88              BBC      #DEV$V_TRM,RAB$L_CTX(R1),80$  ; Exit if input not a terminal
                              001F        89
                              001F        90       ;
                              001F        91       ; Process /[NO]LOGOUT and /[NO]CONTINUE qualifiers.
                              001F        92       ;
    56  00001823 8F   D0    001F        93              MOVL     #IO$_SETMODE!IO$M_TT_CONNECT!- ; Assume /LOGOUT
                              0026        94                       IO$M_TT_DISCON,R6             ;
               57   D4    0026        95              CLRL     R7                             ; Assume /NOCONTINUE
            FFD5'   30    0028        96       10$:   BSBW     DCL$GETDVAL                    ; Get first token
        55   00   D1    002B        97              CMPL     #PTR_K_COMDQUAL,R5             ; Qualifier?
               2F   12    002E        98              BNEQ     20$                            ; No, then parameter
            FFCD'   30    0030        99              BSBW     DCL$GETNVAL                    ; Get qualifier number
    51  00000000'8F   D1    0033       100              CMPL     #CLI$K_CONN_CONT,R1           ; Was it /CONTINUE?
               10   12    003A       101              BNEQ     15$                            ; No, then branch
                              003C       102
               57   D4    003C       103              CLRL     R7                             ; Assume /NOCONTINUE
            E7 53   E8    003E       104              BLBS     R3,10$                         ; Branch if /NOCONTINUE
               57   D6    0041       105              INCL     R7                             ; Set /CONTINUE
    56  00001000 8F   CA    0043       106              BICL     #IO$M_TT_DISCON,R6            ; Set /NOLOGOUT
               DC   11    004A       107              BRB      10$                            ; Get next token
                              004C       108
    56  00001000 8F   C8    004C       109       15$:   BISL     #IO$M_TT_DISCON,R6            ; Assume /LOGOUT
            D2 53   E9    0053       110              BLBC     R3,10$                         ; Branch if /LOGOUT
    56  00001000 8F   CA    0056       111              BICL     #IO$M_TT_DISCON,R6            ; Set /NOLOGOUT
               C9   11    005D       112              BRB      10$                            ; Get next token
                              005F       113
                              005F       114       ;
                              005F       115       ; If /NOLOGOUT, check that terminal is disconnectable.
                              005F       116       ;
           53   51   7D    005F       117       20$:   MOVQ     R1,R3                          ; Save terminal descriptor
        03 56   0C   E0    0062       118              BBS      #IO$V_TT_DISCON,R6,30$         ; Branch if /LOGOUT
```

```
          00A7    30   0066   119           BSBW    DISCONNECTABLE                          ; Check disconnectability
                       0069   120
                       0069   121 ;
                       0069   122 ; Connect to new terminal and optionally logout the process.
                       0069   123 ;
             79    7C   0069   124 30$:         CLRQ    -(R9)                                   ; Allocate an IOSB
        79   53    7D   006B   125           MOVQ    R3,-(R9)                                ; Save terminal name descriptor
                       006E   126           $QIOW_S  CHAN=PRC_W_INPCHAN(R11),-               ; Execute the QIO
                       006E   127                    IOSB=8(R9),-                            ;
                       006E   128                    FUNC=R6,-                               ;
                       006E   129                    EFN=#EXE$C_SYSEFN,-                     ;
                       006E   130                    P1=(R9)                                 ;
        0D 50    E9   008F   131 80$:         BLBC    R0,90$                                  ; Branch if error
  50    08 A9    3C   0092   132           MOVZWL  8(R9),R0                                ; Use IOSB status
        06 50    E9   0096   133           BLBC    R0,90$                                  ; Branch if error
                       0099   134
                       0099   135 ;
                       0099   136 ; Continue current process if requested.
                       0099   137 ;
        03 57    E9   0099   138           BLBC    R7,90$                                  ; Branch if /NOCONTINUE
           0059    31   009C   139           BRW     CONTINUE                                ; Execute continue command
                 05   009F   140 90$:         RSB                                             ; Return
```

CONNECT   L 9
V04-000   - Connect/disconnect process to terminal 15-SEP-1984 23:41:17 VAX/VMS Macro V04-00   Page 5
                          Disconnect process from terminal   4-SEP-1984 23:39:49 [DCL.SRC]CONNECT.MAR;1   (3)

```
                        00A0    142              .SBTTL  Disconnect process from terminal
                        00A0    143  ;+
                        00A0    144  ; DCL$DISCONNECT - Disconnect process from terminal
                        00A0    145  ;
                        00A0    146  ; This routine is called as an internal command to execute the DISCONNECT
                        00A0    147  ; DCL command.
                        00A0    148  ;
                        00A0    149  ; INPUTS:
                        00A0    150  ;
                        00A0    151  ;       R8 = Address of scratch buffer descriptor.
                        00A0    152  ;       R9 = Address of scratch stack.
                        00A0    153  ;       R10 = Base address of command work area.
                        00A0    154  ;       R11 = Base address of process work area.
                        00A0    155  ;
                        00A0    156  ; OUTPUTS:
                        00A0    157  ;
                        00A0    158  ;       The process is disconnected from the terminal.
                        00A0    159  ;-
                        00A0    160
                        00A0    161  DCL$DISCONNECT::                                  ; Disconnect process
                        00A0    162  ;
                        00A0    163  ; Exit if batch job or input is not from a terminal.
                        00A0    164  ;
                        00A0    165              STATUS  NOTDISCON                     ; Assume not disconnectable
48 68 AB   06    E0     00A7    166              BBS     #PRC_V_MODE,PRC_W_FLAGS(R11),30$; Exit if batch
   51    08 AB   D0     00AC    167              MOVL    PRC_L_INPRAB(R11),R1          ; Get addr of input RAB
42 18 A1   02    E1     00B0    168              BBC     #DEV$V_TRM,RAB$L_CTX(R1),30$  ; Exit if input not a terminal
                        00B5    169
                        00B5    170  ;
                        00B5    171  ; Check that terminal is disconnectable.
                        00B5    172  ;
          0058    30    00B5    173              BSBW    DISCONNECTABLE                ; Check disconnectability
                        00B8    174
                        00B8    175  ;
                        00B8    176  ; Disconnect the terminal.
                        00B8    177  ;
            79    7C    00B8    178  10$:         CLRQ    -(R9)                         ; Allocate an IOSB
                        00BA    179              $QIOW_S CHAN=PRC_W_INPCHAN(R11),-     ; Execute the QIO
                        00BA    180                      IOSB=(R9),-                   ;
                        00BA    181                      EFN=#EXE$C_SYSEFN,-           ;
                        00BA    182                      FUNC=#IO$_SETMODE!IO$M_TT_DISCON;
         18 50   E9     00DC    183              BLBC    R0,30$                        ; Branch if error
      50 69     3C     00DF    184              MOVZWL  (R9),R0                       ; Use IOSB status
      12 50     E9     00E2    185              BLBC    R0,30$                        ; Branch if error
                        00E5    186
                        00E5    187  ;
                        00E5    188  ; Process /[NO]CONTINUE qualifier.
                        00E5    189  ;
       FF18'    30     00E5    190              BSBW    DCL$GETDVAL                   ; Get first token
    55   00    D1     00E8    191              CMPL    #PTR_K_COMDQUAL,R5            ; Qualifier?
         03    12     00EB    192              BNEQ    20$                           ; No, then done
      08 53    E9     00ED    193              BLBC    R3,CONTINUE                   ; Branch if /CONTINUE
                        00F0    194
                        00F0    195  20$:         STATUS  NORMAL                        ; Set success status
            05     00F7    196  30$:             RSB                                   ; Return
```

CONNECT   M 9
V04-000   - Connect/disconnect process to terminal 15-SEP-1984 23:41:17   VAX/VMS Macro V04-00   Page   6
                          Contine process after disconnect   4-SEP-1984 23:39:49   [DCL.SRC]CONNECT.MAR;1   (4)

```
                              00F8   198              .SBTTL  Contine process after disconnect
                              00F8   199  ;+
                              00F8   200  ; CONTINUE - Contine process after disconnect
                              00F8   201  ;
                              00F8   202  ; This routine is called to issue the CONTINUE command.
                              00F8   203  ;
                              00F8   204  ; INPUTS:
                              00F8   205  ;
                              00F8   206  ;         R8 = Address of scratch buffer descriptor.
                              00F8   207  ;         R9 = Address of scratch stack.
                              00F8   208  ;         R10 = Base address of command work area.
                              00F8   209  ;         R11 = Base address of process work area.
                              00F8   210  ;
                              00F8   211  ; OUTPUTS:
                              00F8   212  ;
                              00F8   213  ;         The process continues execution.
                              00F8   214  ;-
                              00F8   215
                              00F8   216  CONTINUE:                                    ; Continue process
       51    FF04 CF    9E    00F8   217              MOVAB   CONT,R1                  ; Get addr of continue command
       50      81      9A    00FD   218              MOVZBL  (R1)+,R0                 ; Get descr of command
F896 CA  61      50      28    0100   219              MOVC    R0,(R1),WRK_G_INPBUF(R10)  ; Move command to input buffer
       F895 CA          9E    0106   220              MOVAB   WRK_G_INPBUF-1(R10),-    ; Set input buffer pointer
       F48E CA                010A   221                      WRK_L_CHARPTR(R10)       ;
            FEF0'    31    010D   222              BRW     DCL$CMDSTART             ; Execute the command
```

CONNECT
VO4-000

N 9
- Connect/disconnect process to terminal 15-SEP-1984 23:41:17 VAX/VMS Macro VO4-00   Page 7
Check that terminal is disconnectable   4-SEP-1984 23:39:49 [DCL.SRC]CONNECT.MAR;1   (5)

```
                        0110   224              .SBTTL  Check that terminal is disconnectable
                        0110   225      ;+
                        0110   226      ; DISCONNECTABLE - Check that terminal is disconnectable
                        0110   227      ;
                        0110   228      ; This routine is called to check that the terminal is disconnectable.
                        0110   229      ;
                        0110   230      ; INPUTS:
                        0110   231      ;
                        0110   232      ;       R8 = Address of scratch buffer descriptor.
                        0110   233      ;       R9 = Address of scratch stack.
                        0110   234      ;       R10 = Base address of command work area.
                        0110   235      ;       R11 = Base address of process work area.
                        0110   236      ;
                        0110   237      ; OUTPUTS:
                        0110   238      ;
                        0110   239      ;       Routine only returns to caller if terminal is disconnectable.
                        0110   240      ;-
                        0110   241      DISCONNECTABLE:                                    ; Check disconnectability
              79  D4     0110   242              CLRL    -(R9)                              ; Allocate a return buffer
              79  7C     0112   243              CLRQ    -(R9)                              ; Create the item list
      79  08  A9  DE     0114   244              MOVAL   8(R9),-(R9)                        ; Set the buffer address
79  001C0004  8F  D0     0118   245              MOVL    #DVI$_DEVDEPEND2@16!4,-(R9)         ; Set item code and length
              79  7C     011F   246              CLRQ    -(R9)                              ; Allocate an IOSB
                        0121   247              $GETDVIW_S  CHAN=PRC_W_INPCHAN(R11),-        ; Get the terminal chars
                        0121   248                      ITMLST=8(R9),-                     ;
                        0121   249                      IOSB=(R9),-                        ;
                        0121   250                      EFN=#EXE$C_SYSEFN                  ;
        13  50  E9       013D   251              BLBC    R0,90$                             ; Branch if error
        50  69  3C       0140   252              MOVZWL  (R9),R0                            ; Use IOSB status
        0D  50  E9       0143   253              BLBC    R0,90$                             ; Branch if error
                        0146   254              STATUS  NOTDISCON                          ; Assume not disconnectable
01  18  A9  11  E1       014D   255              BBC     #TT2$V_DISCONNECT,24(R9),90$       ; Branch if not disconnectable
              05         0152   256              RSB                                        ; Return
                        0153   257
              8E  D5     0153   258      90$:    TSTL    (SP)+                              ; Pop return address
              05         0155   259              RSB                                        ; Return status
                        0156   260
                        0156   261              .END
```

B 10

CONNECT   - Connect/disconnect process to terminal 15-SEP-1984 23:41:17 VAX/VMS Macro V04-00   Page 8
Symbol table   4-SEP-1984 23:39:49 [DCL.SRC]CONNECT.MAR;1   (5)

$ST1	= 00000001			PRC_L_SAVAP	00000000	
CLI$K_CONN_CONT	********	X	02	PRC_L_SAVFP	00000004	
CLI$_NORMAL	= 00030001			PRC_L_SEVERITY	00000050	
CLI$_NOTDISCON	= 0003892A			PRC_L_SPWN	000000C0	
CONT	00000000	R	02	PRC_L_STACKLM	000000A4	
CONTINUE	000000F8	R	02	PRC_L_STACKPT	000000A0	
DCL$CMDSTART	********	X	02	PRC_L_STATUS	00000054	
DCL$CONNECT	0000000A	RG	02	PRC_L_STS	00000084	
DCL$DISCONNECT	000000A0	RG	02	PRC_L_STV	00000088	
DCL$GETDVAL	********	X	02	PRC_L_SYMBOL	00000060	
DCL$GETNVAL	********	X	02	PRC_L_TMBX	00000074	
DEV$V_TRM	= 00000002			PRC_L_TRMLIST	00000010	
DISCONNECTABLE	00000110	R	02	PRC_Q_ALLOCREG	00000020	
DVI$_DEVDEPEND2	= 0000001C			PRC_Q_COMMAND	000000E0	
EXE$C_SYSEFN	********	X	02	PRC_Q_FLUSHTIME	000000D0	
IO$M_TT_CONNECT	= 00000800			PRC_Q_GLOBAL	00000028	
IO$M_TT_DISCON	= 00001000			PRC_Q_IMAGENAME	000000D8	
IO$V_TT_DISCON	= 0000000C			PRC_Q_KEYPAD	00000040	
IO$_SETMODE	= 00000023			PRC_Q_LABEL	00000030	
PRC_B_CONTINUE	000000F3			PRC_Q_LOCAL	00000038	
PRC_B_DEFRADIX	000000AE			PRC_Q_SAVEPRIV	000000E8	
PRC_B_EXMDEPMOD	000000AD			PRC_T_OUTDVI	0000011C	
PRC_B_EXMDEPWID	000000AC			PRC_V_MODE	= 00000006	
PRC_B_EXONLYL	0000012D			PRC_W_ASTIOSB	000000C6	
PRC_B_FLAGS2	000000AF			PRC_W_ASTRETN	000000C8	
PRC_B_IMGFLAG	00000078			PRC_W_ASTSTATUS	000000C4	
PRC_B_OUTFLAGS	0000012C			PRC_W_ATTMBX	0000007A	
PRC_B_PROMPTLEN	000000F0			PRC_W_FLAGS	00000068	
PRC_C_LENGTH	00000534			PRC_W_INPCHAN	00000064	
PRC_G_COMMANDS	00000133			PRC_W_ONLEVEL	0000006A	
PRC_G_PROMPT	000000F4			PRC_W_OUTIFI	00000114	
PRC_K_LENGTH	00000534			PRC_W_OUTISI	00000116	
PRC_L_CURRKEY	00000048			PRC_W_OUTMBXCHN	000000CA	
PRC_L_EXMDEPADR	000000A8			PRC_W_OUTMBXREF	000000CE	
PRC_L_EXTARG	00000094			PRC_W_OUTMBXSIZ	000000CC	
PRC_L_EXTBLK	0000008C			PRC_W_PMPTCTRL	000000F1	
PRC_L_EXTCOD	0000009C			PRC_W_WAITIOSB	00000066	
PRC_L_EXTHND	00000090			PTR_B_LEVEL	00000004	
PRC_L_EXTPRM	00000098			PTR_B_NUMBER	00000005	
PRC_L_IDFLNK	000000BC			PTR_B_PARMCNT	00000006	
PRC_L_IMGACTSTS	00000080			PTR_B_VALUE	00000000	
PRC_L_INDCLOCK	0000007C			PTR_C_LENGTH	0000000C	
PRC_L_INDEPTH	0000005C			PTR_K_COMDQUAL	= 00000000	
PRC_L_INDFAB	0000001C			PTR_K_LENGTH	0000000C	
PRC_L_INDINPRAB	00000014			PTR_L_DESCR	00000000	
PRC_L_INDOUTRAB	00000018			PTR_L_ENTITY	00000008	
PRC_L_INPRAB	00000008			RAB$L_CTX	= 00000018	
PRC_L_LASTKEY	0000004C			SYS$GETDVIW	********	GX	02
PRC_L_LSTSTATUS	000000B0			SYS$QIOW	********	GX	02
PRC_L_ONCTLY	000000B8			TT2$V_DISCONNECT	= 00000011	
PRC_L_ONERROR	0000006C			WRK_B_CMDOPT	FFFFFFC3	
PRC_L_OUTOFBAND	000000B4			WRK_B_MAXPARM	FFFFFFD0	
PRC_L_OUTRAB	0000000C			WRK_B_MINPARM	FFFFFFD1	
PRC_L_OUTRABCTX	00000118			WRK_B_PARMCNT	FFFFFFCE	
PRC_L_PPFLIST	00000070			WRK_B_PARMSUM	FFFFFFCF	
PRC_L_RECALLPTR	0000012F			WRK_B_RECALLCNT	FFFFFFC5	
PRC_L_RESTART	00000058			WRK_B_VALLEV	FFFFFFC4	

```
WRK_B_VERBTYP              FFFFFFC2
WRK_C_LENGTH               FFFFF486
WRK_G_BUFFER               FFFFF492
WRK_G_INPBUF               FFFFF896
WRK_G_RESULT               FFFFF9B6
WRK_K_LENGTH               FFFFF486
WRK_L_CHARPTR              FFFFF48E
WRK_L_DISALLOW             FFFFFFE6
WRK_L_ERRORRTN             FFFFF9AE
WRK_L_EXPANDPTR            FFFFF486
WRK_L_IMAGE                FFFFFFE2
WRK_L_MARKPTR              FFFFF48A
WRK_L_PAROUT               FFFFFFD2
WRK_L_PMPTADDR             FFFFF9A2
WRK_L_PROMPTRTN            FFFFF9A6
WRK_L_PROPTR               FFFFFFC6
WRK_L_QUABLK               FFFFFFCA
WRK_L_READRTN              FFFFF9AA
WRK_L_RECALLPTR            FFFFFFEA
WRK_L_RSLEND               FFFFFFB6
WRK_L_RSLNXT               FFFFFFBA
WRK_L_SAVAP                FFFFFFF8
WRK_L_SAVFP                FFFFFFFC
WRK_L_SAVSP                FFFFFFF4
WRK_L_SIGNALRTN            FFFFFFD6
WRK_L_SPECRTN              FFFFF9B2
WRK_L_TAB_VEC              FFFFFFDE
WRK_L_VERB                 FFFFFFBE
WRK_W_FLAGS                FFFFFFF0
WRK_W_FLAGS2               FFFFFFF2
WRK_W_IMGCHAN              FFFFFFEE
WRK_W_PMPTLEN              FFFFF99E
```

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

PSECT name	Allocation		PSECT No.		Attributes											
. ABS .	00000000	(0.)	00	(0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
ABS	FFFFFFFC	(0.)	01	(1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
DCL$ZCODE	00000156	(342.)	02	(2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	15	00:00:00.09	00:00:00.82
Command processing	106	00:00:00.67	00:00:05.46
Pass 1	310	00:00:12.27	00:00:35.80
Symbol table sort	0	00:00:01.76	00:00:05.20
Pass 2	48	00:00:01.96	00:00:06.83
Symbol table output	17	00:00:00.14	00:00:00.37
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00

Assembler run totals   498   00:00:16.91   00:00:54.50

The working set limit was 1200 pages.
63550 bytes (125 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1219 non-local and 10 local symbols.
261 source lines were read in Pass 1, producing 14 object records in Pass 2.
40 pages of virtual memory were used to define 26 macros.

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

```
Macro library name                        Macros defined
-------------------                       ---------------
_$255$DUA28:[SYSLIB]SYSBLDMLB.MLB;1              0
_$255$DUA28:[DCL.OBJ]DCL.MLB;1                   6
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                   0
_$255$DUA28:[SYSLIB]STARLET.MLB;2               14
TOTALS (all libraries)                          20
```

1420 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

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

INTDEF
SDL

COMMAND
LIS

DESCRVAL
LIS

CONVERT
LIS

CLIMAC
MAR

CLIMSG
LIS

CONNECT
LIS

DCLPARSE
LIS

CHARMANIP
LIS

EXAMDEP
LIS

EXIT
LIS

INTIMAGES
MAR

DCXSTART
LIS

CLIGBL
LIS

DISALLOW
LIS

CLINT
LIS

CANCEL
LIS