M 8

```
DDDDDDDDDDDD     TTTTTTTTTTTTTTT     SSSSSSSSSSSS  DDDDDDDDDDDD     TTTTTTTTTTTTTTT  RRRRRRRRRRRR
DDDDDDDDDDDD     TTTTTTTTTTTTTTT     SSSSSSSSSSSS  DDDDDDDDDDDD     TTTTTTTTTTTTTTT  RRRRRRRRRRRR
DDDDDDDDDDDD     TTTTTTTTTTTTTTT     SSSSSSSSSSSS  DDDDDDDDDDDD     TTTTTTTTTTTTTTT  RRRRRRRRRRRR
DDD         DDD        TTT        SSS              DDD         DDD        TTT        RRR         RRR
DDD         DDD        TTT        SSS              DDD         DDD        TTT        RRR         RRR
DDD         DDD        TTT        SSS              DDD         DDD        TTT        RRR         RRR
DDD         DDD        TTT        SSS              DDD         DDD        TTT        RRR         RRR
DDD         DDD        TTT        SSS              DDD         DDD        TTT        RRR         RRR
DDD         DDD        TTT        SSS              DDD         DDD        TTT        RRR         RRR
DDD         DDD        TTT           SSSSSSSSS     DDD         DDD        TTT        RRRRRRRRRRRR
DDD         DDD        TTT           SSSSSSSSS     DDD         DDD        TTT        RRRRRRRRRRRR
DDD         DDD        TTT           SSSSSSSSS     DDD         DDD        TTT        RRRRRRRRRRRR
DDD         DDD        TTT                    SSS  DDD         DDD        TTT        RRR   RRR
DDD         DDD        TTT                    SSS  DDD         DDD        TTT        RRR   RRR
DDD         DDD        TTT                    SSS  DDD         DDD        TTT        RRR   RRR
DDD         DDD        TTT                    SSS  DDD         DDD        TTT        RRR      RRR
DDD         DDD        TTT                    SSS  DDD         DDD        TTT        RRR      RRR
DDD         DDD        TTT                    SSS  DDD         DDD        TTT        RRR      RRR
DDDDDDDDDDDD           TTT        SSSSSSSSSSSS     DDDDDDDDDDDD           TTT        RRR         RRR
DDDDDDDDDDDD           TTT        SSSSSSSSSSSS     DDDDDDDDDDDD           TTT        RRR         RRR
DDDDDDDDDDDD           TTT        SSSSSSSSSSSS     DDDDDDDDDDDD           TTT        RRR         RRR
```

```
**FILE**ID**DTSPARSE                                         F  5




DDDDDDDD    TTTTTTTTTT    SSSSSSSS  PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
DDDDDDDD    TTTTTTTTTT    SSSSSSSS  PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
DD      DD      TT      SS          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD      TT      SS          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD      TT      SS          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD      TT      SS          PP      PP  AA      AA  RR      RR  SS          EE
DD      DD      TT        SSSSSS    PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
DD      DD      TT        SSSSSS    PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
DD      DD      TT              SS  PP          AAAAAAAAAA  RR  RR              SS  EE
DD      DD      TT              SS  PP          AAAAAAAAAA  RR  RR              SS  EE
DD      DD      TT              SS  PP          AA      AA  RR    RR            SS  EE              ....
DD      DD      TT              SS  PP          AA      AA  RR    RR            SS  EE              ....
DDDDDDDD        TT      SSSSSSSS    PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....
DDDDDDDD        TT      SSSSSSSS    PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....


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

Table of contents

```
    (2)       42        DECLARATIONS
    (3)       60        TST$PARSE - COMMAND PARSE ROUTINE
    (4)      188        PARSE ROUTINE--PARAMETER EVALUATION AND DEFAULTING
    (5)      296        PARSE ROUTINE--QUALIFIER EVALUATION
    (6)      352        PARSE ROUTINE--QUALIFIER VALUE EVALUATION
    (7)      549        TST$NEXTCHAR - EXAMINE NEXT CHARACTER
    (8)      630        TST$MATCH - KEYWORD MATCH ROUTINE
    (9)      688        TST$CVTU_DTB - CONVERT UNSIGNED DECIMAL TO BINARY
```

```
0000            .TITLE  TST$DTSPARSE - PARSE DTS COMMAND LINE
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY: DTS/DTR DECNET TEST PACKAGE
0000    ;
0000    ; ABSTRACT: THIS MODULE PARSES A COMMAND LINE INPUT BY DTS.
0000    ;
0000    ; ENVIRONMENT: DTS/DTR RUN IN USER MODE AND REQUIRE NETWORK PRIVILEGE.
0000    ;
0000    ; AUTHOR: JAMES A. KRYCKA,      CREATION DATE: 11-AUG-77
0000    ;
0000    ; MODIFICATIONS:
0000    ;
0000    ;--
```

```
0000                .SBTTL  DECLARATIONS
0000
0000        ;
0000        ; INCLUDE FILES:
0000        ;
0000                FLGDEF                              ; DEFINE COMMAND PARSE FLAGS
0000                CMDDEF                              ; DEFINE COMMAND LANGUAGE SYMBOLS
0000                VLDDEF                              ; DEFINE VALID QUALIFIER FLAGS
0000                .IIF NE K_LIST_MEB, .LIST MEB       ; DEFINED IN DTPREFIX.MAR
0000        ;
0000        ; EQUATED SYMBOLS:
0000        ;
0000                NONE
0000        ;
0000        ; OWN STORAGE:
0000        ;
0000                NONE
```

```
           0000        .SBTTL  TST$PARSE - COMMAND PARSE ROUTINE
       00000000        .PSECT  TST$CODE        NOWRT
           0000  P::                                   ; SYMBOL FOR DEBUGGING PURPOSES
           0000
           0000  ;++
           0000  ; FUNCTIONAL DESCRIPTION:
           0000  ;
           0000  ;       NONE
           0000  ;
           0000  ; CALLING SEQUENCE:
           0000  ;
           0000  ;       CALL    #0,TST$PARSE
           0000  ;
           0000  ; INPUT PARAMETERS:
           0000  ;
           0000  ;       R8      THE ADDRESS OF THE NEXT CHARACTER IN THE BUFFER
           0000  ;       R9      THE ADDRESS OF THE END OF THE BUFFER + 1
           0000  ;
           0000  ; IMPLICIT INPUTS:
           0000  ;
           0000  ;       NONE
           0000  ;
           0000  ; OUTPUT PARAMETERS:
           0000  ;
           0000  ;       R0-R9   DESTROYED
           0000  ;       R10     COMMAND PARAMETER VALUE (TESTTYPE)
           0000  ;       R11     UPDATED PARSE FLAGS
           0000  ;
           0000  ; IMPLICIT OUTPUTS:
           0000  ;
           0000  ;       TST$GB_BACK
           0000  ;       TST$GB_DISPLAY
           0000  ;       TST$GB_FLOW
           0000  ;       TST$GB_NAK
           0000  ;       TST$GT_NODENAME
           0000  ;       TST$GB_PRINT
           0000  ;       TST$GB_RETURN
           0000  ;       TST$GB_RQUEUE
           0000  ;       TST$GL_SECONDS
           0000  ;       TST$GW_SIZE
           0000  ;       TST$GL_SPEED
           0000  ;       TST$GB_SQUEUE
           0000  ;       TST$GB_TEST
           0000  ;       TST$GB_TYPE
           0000  ;
           0000  ; COMPLETION CODES:
           0000  ;
           0000  ;       NONE
           0000  ;
           0000  ; SIDE EFFECTS:
           0000  ;
           0000  ;       NONE
           0000  ;
           0000  ;--
           0000
     0000  0000        .ENTRY  TST$PARSE,^M<>          ; ENTRY POINT
           0002
```

```
                   0002  ;
                   0002  ; DETERMINE IF THE NEXT SYNTACTICAL ELEMENT OF THE COMMAND LINE IS A
                   0002  ; PARAMETER OR QUALIFIER, OR IF THE END OF THE INPUT LINE HAS BEEN
                   0002  ; REACHED.
                   0002  ;
                   0002
                   0002  NEXT_ELEMENT:                                   ;
      02F7   30    0002          BSBW    TST$NEXTCHAR                    ; GET CHARACTER
                   0005  REEXAMINE_CHAR:
                   0005          $CASEB  SELECTOR=R0,DISPL=<-            ; CHARACTER:
                   0005                          END_OF_LINE-            ;  END-OF-LINE
                   0005                          QUAL-                   ;  SLASH
                   0005                          PARSE_ERROR-            ;  EQUAL_SIGN OR COLON
                   0005                          SPACE_OR_TAB-           ;  SPACE OR TAB
                   0005                          PARAM-                  ;  NONE OF THE ABOVE
                   0005          >                                       ;
                   0013
                   0013  ;
                   0013  ; A SPACE OR TAB HAS BEEN ENCOUNTERED. THIS IMPLIES THAT A QUALIFIER (/)
                   0013  ; CAN NOT IMMEDIATELY FOLLOW. ANOTHER SPACE OR TAB, A PARAMETER, OR A
                   0013  ; QUALIFIER MAY FOLLOW.
                   0013  ;
                   0013
                   0013  SPACE_OR_TAB:                                   ;
   5B   08   88    0013          BISB2   #FLG_M_DELIMITER,R11            ; SET DELIMITER FLAG
        EA   11    0016          BRB     NEXT_ELEMENT                    ; CONTINUE
                   0018
                   0018  ;
                   0018  ; A QUALIFIER FOLLOWS. DISCARD THE SLASH DELIMITER AND PROCEED.
                   0018  ;
                   0018
                   0018  QUAL:                                           ;
1E 5B   03   E0    0018          BBS     #FLG_V_DELIMITER,R11,-          ; SLASH CANNOT FOLLOW
                   001C                  PARSE_ERROR                     ;  SPACE OR TAB
      00FD   30    001C          BSBW    QUALIFIER                       ; PROCESS QUALIFIER
        E4   11    001F          BRB     REEXAMINE_CHAR                  ; PROCESS QUALIFIER DELIMITER
                   0021
                   0021  ;
                   0021  ; A PARAMETER FOLLOWS. THE COMMAND CAN HAVE ONLY ONE PARAMETER.
                   0021  ;
                   0021
                   0021  PARAM:                                          ;
15 5B   02   E2    0021          BBSS    #FLG_V_PARAMETER,R11,-          ; ERROR IF NOT FIRST PARAMETER
                   0025                  PARSE_ERROR                     ;
   5B   08   8A    0025          BICB2   #FLG_M_DELIMITER,R11            ; CLEAR DELIMITER FLAG
        14   10    0028          BSBB    PARAMETER                       ; PROCESS PARAMETER
        D9   11    002A          BRB     REEXAMINE_CHAR                  ; PROCESS PARAMETER DELIMITER
                   002C
                   002C  ;
                   002C  ; THE END OF THE LINE HAS BEEN REACHED. SET FLAG IF THE COMMAND LINE
                   002C  ; REQUIRES ANOTHER LINE OF INPUT; OTHERWISE, DETERMINE WHETHER THE
                   002C  ; REQUIRED PARAMETER HAS BEEN RECEIVED.
                   002C  ;
                   002C
                   002C  END_OF_LINE:                                    ;
   2D   51   91    002C          CMPB    R1,#^A/-/                       ; IS COMMAND LINE CONTINUED?
        05   13    002F          BEQLU   10$                             ; YES
```

```
05 5B    02   E5  0031             BBCC    #FLG_V_PARAMETER,R11,-  ; IT IS AN ERROR IF NO
                  0035                     PARSE_ERROR             ;  PARAMETER HAS BEEN PROCESSED
              04  0035             RET                             ; EXIT TO DTSMAIN
   5B    02   88  0036        10$: BISB2   #FLG_M_MULTILINE,R11    ; SET CONTINUATION FLAG
              04  0039             RET                             ; EXIT TO DTSMAIN
                  003A
                  003A        ;
                  003A        ; AN ERROR HAS BEEN ENCOUNTERED DURING COMMAND LINE PARSING.
                  003A        ;
                  003A
                  003A        PARSE_ERROR:                         ; CONTROL POINT
   5B    01   88  003A             BISB2   #FLG_M_PARSERROR,R11    ; SET PARSE ERROR FLAG
              04  003D             RET                             ; EXIT TO DTSMAIN
```

```
                              003E          .SBTTL  PARSE ROUTINE--PARAMETER EVALUATION AND DEFAULTING
                              003E
                              003E  ;+
                              003E  ; PARAMETER IS A SPECIAL PURPOSE SUBROUTINE TO PARSE THE COMMAND PARAMETER
                              003E  ; AND TO APPLY PARAMETER QUALIFIER DEFUALTS. THE PARAMETER STRING IS
                              003E  ; STORED IN TST$GT_KEYWORD.
                              003E  ;-
                              003E
                              003E  PARAMETER:                                        ; CONTROL POINT
           52   0000'CF   DE  003E          MOVAL   W^TST$GT_KEYWORD,R2       ; GET ADDRESS OF BUFFER
    62   20202020 8F      D0  0043          MOVL    #^A/    7,(R2)            ; FILL KEYWORD STRING WITH SPACES
                     53   D4  004A          CLRL    R3                        ; ZERO CHARACTER COUNT
                     11   11  004C          BRB     PARAM_CHAR                ; STORE FIRST CHARACTER
                              004E  PARAM_LOOP:
                   02AB   30  004E          BSBW    TST$NEXTCHAR              ; GET NEXT CHARACTER
                              0051          $CASEB  SELECTOR=R0,DISPL=<-      ; CHARACTER:
                              0051                          PARAM_DELIMITER-  ;  END-OF-LINE
                              0051                          PARAM_DELIMITER-  ;  SLASH
                              0051                          PARSE_ERROR-      ;  EQUAL_SIGN OR COLON
                              0051                          PARAM_DELIMITER-  ;  SPACE OR TAB
                              0051                          PARAM_CHAR-       ;  NONE OF THE ABOVE
                              0051                  >                         ;
                              005F  PARAM_CHAR:                               ;
              04     53   D1  005F          CMPL    R3,#4                     ; STORE ONLY FIRST 4 CHARACTERS
                     EA   13  0062          BEQLU   PARAM_LOOP                ; IGNORE THIS CHARACTER
              82     51   90  0064          MOVB    R1,(R2)+                  ; STORE CHARACTER
                     53   D6  0067          INCL    R3                        ; INCREMENT CHARACTER COUNT
                     E3   11  0069          BRB     PARAM_LOOP                ; CONTINUE
                              006B  PARAM_DELIMITER:                          ;
           54   0000'CF   DE  006B          MOVAL   W^TST$AZ_PARAM,R4         ; GET ADDRESS OF KEYWORD TABLE
           56   0000'CF   DE  0070          MOVAL   W^TST$GT_KEYWORD,R6       ; GET ADDRESS OF STRING TO MATCH
                   02BC   30  0075          BSBW    (ST$MATCH                 ; FIND TABLE INDEX OF KEYWORD
           0000'CF   55   F6  0078          CVTLB   R5,W^TST$GB_TEST          ; UPDATE TESTTYPE
              5A     55   D0  007D          MOVL    R5,R10                    ; SAVE IN R10 AS RETURN VALUE!!!
                              0080
                              0080  ;
                              0080  ; APPLY PARAMETER QUALIFIER DEFAULTS (NOT COMMAND QUALIFIER DEFAULTS)
                              0080  ; AND DENOTE VALID (PERMITTED) QUALIFIIERS FOR THE COMMAND.
                              0080  ;
                              0080
                              0080          $CASEB  SELECTOR=R10,DISPL=<-     ; TEST:
                              0080                          CONNTEST-         ;  CONNECT TEST
                              0080                          DATATEST-         ;  DATA TEST
                              0080                          DISCTEST-         ;  DISCONNECT TEST
                              0080                          INTETEST-         ;  INTERRUPT TEST
                              0080                          MISCTEST-         ;  MISCELLANEOUS TEST
                              0080                  >                         ;
                              008E  CONNTEST:                                         ; DEFAULTS FOR:
           0000'CF   00   90  008E          MOVB    #DFT_K_RETURN_CO,W^TST$GB_RETURN ; RETURN QUALIFIER
           0000'CF   01   90  0093          MOVB    #DFT_K_TYPE_CO,W^TST$GB_TYPE     ; TYPE QUALIFIER
                          C8  0098          BISL2   #VLD_M_NORETURN-          ; DENOTE VALID QUALIFIERS:
                              0099                  !VLD_M_RETURN-            ;
                              0099                  !VLD_M_TYPE,-             ;
 0000'CF    00409000 8F           0099                  W^TST$GL_VALID            ;
                          05  00A1          RSB                               ; EXIT
                              00A2  DATATEST:                                         ; DEFAULTS FOR:
           0000'CF   00   90  00A2          MOVB    #DFT_K_BACK,W^TST$GB_BACK         ; BACK PRESSURE CONTROL
```

```
              0000'CF    02   90  00A7   245              MOVB     #DFT_K_FLOW,W^TST$GB_FLOW        ; FLOW CONTROL
              0000'CF    00   90  00AC   246              MOVB     #DFT_K_NAK,W^TST$GB_NAK          ; NAK CONTROL
              0000'CF    01   90  00B1   247              MOVB     #DFT_K_RQUEUE_DA,W^TST$GB_RQUEUE ; DTR QUEUE COUNT
         0000'CF    0080 8F   B0  00B6   248              MOVW     #DFT_K_SIZE_DA,W^TST$GW_SIZE     ; MESSAGE SIZE
              0000'CF    01   90  00BD   249              MOVB     #DFT_K_SQUEUE_DA,W^TST$GB_SQUEUE ; DTS QUEUE COUNT
              0000'CF    1E   D0  00C2   250              MOVL     #DFT_K_TIME_DA,W^TST$GL_SECONDS  ; DATA TEST DURATION
              0000'CF    00   90  00C7   251              MOVB     #DFT_K_TYPE_DA,W^TST$GB_TYPE     ; TYPE QUALIFIER
                              C8  00CC   252              BISL2    #VLD_M_BACK-               ; DENOTE VALID QUALIFIERS:
                                  00CD   253                       !VLD_M_FLOW-               ;
                                  00CD   254                       !VLD_M_HOURS-              ;
                                  00CD   255                       !VLD_M_MINUTES-            ;
                                  00CD   256                       !VLD_M_NAK-                ;
                                  00CD   257                       !VLD_M_NOBACK-             ;
                                  00CD   258                       !VLD_M_NOFLOW-             ;
                                  00CD   259                       !VLD_M_NONAK-              ;
                                  00CD   260                       !VLD_M_RQUEUE-             ;
                                  00CD   261                       !VLD_M_SECONDS-            ;
                                  00CD   262                       !VLD_M_SIZE-               ;
                                  00CD   263                       !VLD_M_SQUEUE-             ;
                                  00CD   264                       !VLD_M_TYPE,-              ;
     0000'CF    0057067D 8F           00CD   265                       W^TST$GL_VALID             ;
                              05  00D5   266                       RSB                        ; EXIT
                                  00D6   267 DISCTEST:                                              ; DEFAULTS FOR:
              0000'CF    00   90  00D6   268              MOVB     #DFT_K_RETURN_DI,W^TST$GB_RETURN ; RETURN QUALIFIER
              0000'CF    01   90  00DB   269              MOVB     #DFT_K_TYPE_DI,W^TST$GB_TYPE     ; TYPE QUALIFIER
                              C8  00E0   270              BISL2    #VLD_M_NORETURN-           ; DENOTE VALID QUALIFIERS:
                                  00E1   271                       !VLD_M_RETURN-             ;
                                  00E1   272                       !VLD_M_TYPE,-              ;
     0000'CF    00409000 8F           00E1   273                       W^TST$GL_VALID             ;
                              05  00E9   274                       RSB                        ; EXIT
                                  00EA   275 INTETEST:                                              ; DEFAULTS FOR:
              0000'CF    01   90  00EA   276              MOVB     #DFT_K_RQUEUE_IN,W^TST$GB_RQUEUE ; DTR QUEUE COUNT
              0000'CF    10   B0  00EF   277              MOVW     #DFT_K_SIZE_IN,W^TST$GW_SIZE     ; MESSAGE SIZE
              0000'CF    01   90  00F4   278              MOVB     #DFT_K_SQUEUE_IN,W^TST$GB_SQUEUE ; DTS QUEUE COUNT
              0000'CF    1E   D0  00F9   279              MOVL     #DFT_K_TIME_IN,W^TST$GL_SECONDS  ; INTERRUPT TEST DURATION
              0000'CF    00   90  00FE   280              MOVB     #DFT_K_TYPE_IN,W^TST$GB_TYPE     ; TYPE QUALIFIER
                              C8  0103   281              BISL2    #VLD_M_HOURS-              ; DENOTE VALID QUALIFIERS:
                                  0104   282                       !VLD_M_MINUTES-            ;
                                  0104   283                       !VLD_M_RQUEUE-             ;
                                  0104   284                       !VLD_M_SECONDS-            ;
                                  0104   285                       !VLD_M_SIZE-               ;
                                  0104   286                       !VLD_M_SQUEUE-             ;
                                  0104   287                       !VLD_M_TYPE,-              ;
     0000'CF    00570018 8F           0104   288                       W^TST$GL_VALID             ;
                              05  010C   289                       RSB                        ; EXIT
                                  010D   290 MISCTEST:                                              ; DEFAULTS FOR:
              0000'CF    00   90  010D   291              MOVB     #DFT_K_TYPE_MI,W^TST$GB_TYPE     ; TYPE QUALIFIER
               00400000 8F   C8  0112   292              BISL2    #VLD_M_TYPE,-              ; DENOTE VALID QUALIFIERS:
                    0000'CF       0118   293                       W^TST$GL_VALID             ;
                              05  011B   294              RSB                                 ; EXIT
```

```
                        011C                 .SBTTL  PARSE ROUTINE--QUALIFIER EVALUATION
                        011C      ;+
                        011C      ; QUALIFIER IS A SPECIAL PURPOSE SUBROUTINE TO PARSE A COMMAND QUALIFIER OR
                        011C      ; A PARAMETER QUALIFIER. THE QUALIFIER STRING IS STORED IN TST$GT_KEYWORD
                        011C      ; AND THE ASSOCIATED QUALIFIER VALUE (IF ANY) IS STORED IN TST$GT_VALUE.
                        011C      ;-
                        011C
                        011C      QUALIFIER:                                      ; CONTROL POINT
                        011C                 FILLBUF DST=W^TST$GT_KEYWORD-        ; FILL KEYWORD AND QUALIFIER
                        011C                         SIZE=#12-                    ;  VALUE STRINGS WITH SPACES
                        011C                         CHAR=<#^A/ />                ;  NOTE R0-R5 ARE DESTROYED!
 52   0000'CF      DE   0125                 MOVAL   W^TST$GT_KEYWORD,R2          ; GET ADDRESS OF BUFFER
           53      D4   012A                 CLRL    R3                           ; ZERO CHARACTER COUNT
                        012C      QUAL_LOOP:                                      ;
        01CD       30   012C                 BSBW    TST$NEXTCHAR                 ; GET NEXT CHARACTER
                        012F      QUAL_REEXAMINE:                                 ;
                        012F                 $CASEB  SELECTOR=R0,DISPL=<-         ; CHARACTER:
                        012F                                 QUAL_DELIMITER-      ;  END-OF-LINE
                        012F                                 QUAL_DELIMITER-      ;  SLASH
                        012F                                 QUAL_VALUE-          ;  EQUAL_SIGN OR COLON
                        012F                                 QUAL_DELIMITER-      ;  SPACE OR TAB
                        012F                                 QUAL_CHAR-           ;  NONE OF THE ABOVE
                        012F                 >                                    ;
                        013D      QUAL_VALUE:                                     ; DISCARD THE EQUAL_SIGN OR COLON
 52   0000'CF      DE   013D                 MOVAL   W^TST$GT_VALUE,R2            ; GET ADDRESS OF QUALIFIER VALUE
           53      D4   0142                 CLRL    R3                           ; ZERO CHARACTER COUNT
                        0144      VALUE_LOOP:                                     ;
        01B5       30   0144                 BSBW    TST$NEXTCHAR                 ; GET NEXT CHARACTER
                        0147                 $CASEB  SELECTOR=R0,DISPL=<-         ; CHARACTER:
                        0147                                 QUAL_REEXAMINE-      ;  END-OF-LINE
                        0147                                 QUAL_REEXAMINE-      ;  SLASH
                        0147                                 VALUE_CHAR-          ;  EQUAL_SIGN OR COLON
                        0147                                 QUAL_REEXAMINE-      ;  SPACE OR TAB
                        0147                                 VALUE_CHAR-          ;  NONE OF THE ABOVE
                        0147                 >                                    ;
                        0155      VALUE_CHAR:                                     ;
      08   53      D1   0155                 CMPL    R3,#8                        ; STORE ONLY FIRST 8 CHARACTERS
           EA      13   0158                 BEQLU   VALUE_LOOP                   ; IGNORE THIS CHARACTER
      82   51      90   015A                 MOVB    R1,(R2)+                     ; STORE CHARACTER
           53      D6   015D                 INCL    R3                           ; INCREMENT CHARACTER COUNT
           E3      11   015F                 BRB     VALUE_LOOP                   ; CONTINUE
                        0161      QUAL_CHAR:                                      ;
      04   53      D1   0161                 CMPL    R3,#4                        ; STORE ONLY FIRST 4 CHARACTERS
           C6      13   0164                 BEQLU   QUAL_LOOP                    ; IGNORE THIS CHARACTER
      82   51      90   0166                 MOVB    R1,(R2)+                     ; STORE CHARACTER
           53      D6   0169                 INCL    R3                           ; INCREMENT CHARACTER COUNT
           BF      11   016B                 BRB     QUAL_LOOP                    ; CONTINUE
                        016D      QUAL_DELIMITER:                                 ;
 54   0000'CF      DE   016D                 MOVAL   W^TST$AZ_QUAL,R4             ; GET ADDRESS OF KEYWORD TABLE
 56   0000'CF      DE   0172                 MOVAL   W^TST$GT_KEYWORD,R6          ; GET ADDRESS OF STRING TO MATCH
         01BA      30   0177                 BSBW    TST$MATCH                    ; FIND TABLE INDEX OF KEYWORD
03 0000'CF   55    E0   017A                 BBS     R5,W^TST$GL_VALID,10$        ; IS THIS A VALID QUALIFIER?
         FEB7      31   0180                 BRW     PARSE_ERROR                  ; NO, NOT IN THIS CONTEXT
           01      10   0183      10$:       BSBB    QUAL_DISPATCH                ; GO TO QUALIFIER SPECIFIC CODE
                   05   0185                 RSB                                  ; EXIT
```

```
                         0186          .SBTTL  PARSE ROUTINE--QUALIFIER VALUE EVALUATION
                         0186
                         0186  ;+
                         0186  ; QUAL_DISPATCH IS A SPECIAL PURPOSE SUBROUTINE THAT CONTAINS QUALIFIER
                         0186  ; SPECIFIC CODE. IT EXISTS AS A SUBROUTINE TO UTILIZE ''RSB'' TO RETURN
                         0186  ; FROM A ''CASE'' INSTRUCTION INSTEAD OF USING 'BRW''.
                         0186  ;-
                         0186
                         0186  QUAL_DISPATCH:                                  ; CONTROL POINT
 56    0000'CF      DE   0186          MOVAL   W^TSTSGT_VALUE,R6               ; GET ADDRESS OF QUALIFIER VALUE
                         018B                                                  ;  STRING FOR POSSIBLE USE BY TSTSMATCH
                         018B          $CASEB  SELECTOR=R5,DISPL=<-            ; DISPATCH TO APPROPRIATE CODE
                         018B                          BACK-                   ;  BACK PRESSURE CONTROL
                         018B                          DISPLAY-                ;  DISPLAY EACH MESSAGE
                         018B                          FLOW-                   ;  FLOW CONTROL
                         018B                          HOURS-                  ;  TIME OF TEST IN HOURS
                         018B                          MINUTES-                ;  TIME OF TEST IN MINUTES
                         018B                          NAK-                    ;  NAK CONTROL
                         018B                          NOBACK-                 ;  NO BACK PRESSURE CONTROL
                         018B                          NODENAME-               ;  NODENAME
                         018B                          NODISPLAY-              ;  DO NOT DISPLAY EACH MESSAGE
                         018B                          NOFLOW-                 ;  NO FLOW CONTROL
                         018B                          NONAK-                  ;  NO NAK CONTROL
                         018B                          NOPRINT-                ;  NO PRINT OPTION FOR DTR
                         018B                          NORETURN-               ;  NO USERDATA TO RETURN
                         018B                          NOSTATISTICS-           ;  NO STATISTICS DESIRED
                         018B                          PRINT-                  ;  PRINT OPTION FOR DTR
                         018B                          RETURN-                 ;  RETURN USERDATA
                         018B                          RQUEUE-                 ;  DTR QUEUE
                         018B                          SECONDS-                ;  TIME OF TEST IN SECONDS
                         018B                          SIZE-                   ;  MESSAGE SIZE
                         018B                          SPEED-                  ;  SPEED OF COMMUNICATIONS LINE
                         018B                          SQUEUE-                 ;  DTS QUEUE
                         018B                          STATISTICS-             ;  STATISTICS DESIRED
                         018B                          TYPE-                   ;  TEST TYPE (SUBFUNCTION)
                         018B          >                                       ;
                         01BD  BACK:                                           ; PROCESS BACK QUALIFIER
   57    80 8F      9A   01BD          MOVZBL  #MAX_K_BACK,R7                  ; DEFINE MAXIMUM VALUE
          018D      30   01C1          BSBW    TSTSCVTU_DTB                    ; CONVERT DIGITS TO BINARY VALUE
 0000'CF   56       F6   01C4          CVTLB   R6,W^TSTSGB_BACK                ; UPDATE BACK PRESSURE CONTROL
                    05   01C9          RSB                                     ; EXIT
                         01CA
                         01CA  DISPLAY:                                        ; PROCESS DISPLAY QUALIFIER
      57   26       D0   01CA          MOVL    #MAX_K_DISPLAY,R7               ; DEFINE MAXIMUM VALUE
          0181      30   01CD          BSBW    TSTSCVTU_DTB                    ; CONVERT DIGITS TO BINARY VALUE
 0000'CF   56       F6   01D0          CVTLB   R6,W^TSTSGB_DISPLAY             ; UPDATE DISPLAY VALUE
                    05   01D5          RSB                                     ; EXIT
                         01D6
                         01D6  FLOW:                                           ; PROCESS FLOW QUALIFIER VALUE
 54    0000'CF      DE   01D6          MOVAL   W^TSTSAZ_FLOW,R4                ; GET ADDRESS OF KEYWORD TABLE
          0156      30   01DB          BSBW    TSTSMATCH                       ; FIND TABLE INDEX OF KEYWORD
            55      D6   01DE          INCL    R5                              ;
 0000'CF   55       F6   01E0          CVTLB   R5,W^TSTSGB_FLOW                ; UPDATE FLOW CONTROL FIELD
                    05   01E5          RSB                                     ; EXIT
                         01E6
                         01E6  HOURS:                                          ; PROCESS HOURS QUALIFIER VALUE
 7E    0E10 8F      3C   01E6          MOVZWL  #3600,-(SP)                     ; # SECONDS IN 1 HOUR
```

```
    57   64 8F    9A  01EB                  MOVZBL  #<MAX_K_TIME_DA/3600>,R7 ; DEFINE MAXIMUM HOUR VALUE
           008A   31  01EF                  BRW     TIME                     ; BRANCH TO COMMON CODE
                      01F2
                      01F2      MINUTES:                                     ; PROCESS MINUTES QUALIFIER VALUE
        7E   3C   9A  01F2                  MOVZBL  #60,-(SP)                ; # SECONDS IN 1 MINUTE
    57 1770 8F    3C  01F5                  MOVZWL  #<MAX_K_TIME_DA/60>,R7   ; DEFINE MAXIMUM MINUTE VALUE
           007F   31  01FA                  BRW     TIME                     ; BRANCH TO COMMON CODE
                      01FD
                      01FD      NAK:                                         ; PROCESS NAK QUALIFIER
    57   80 8F    9A  01FD                  MOVZBL  #MAX_K_NAK,R7            ; DEFINE MAXIMUM VALUE
           014D   30  0201                  BSBW    TST$CVTU_DTB             ; CONVERT DIGITS TO BINARY VALUE
 0000'CF     56   F6  0204                  CVTLB   R6,W^TST$GB_NAK          ; UPDATE NAK CONTROL
                  05  0209                  RSB                              ; EXIT
                      020A
                      020A      NOBACK:                                      ; PROCESS NOBACK QUALIFIER
                      020A                  ASSUME  VAL_K_BACK_NO,EQ,0       ;
     0000'CF      94  020A                  CLRB    W^TST$GB_BACK            ; UPDATE BACK PRESSURE CONTROL VALUE
                  05  020E                  RSB                              ; EXIT
                      020F
                      020F      NODENAME:                                    ; PROCESS NODENAME QUALIFIER VALUE
                      020F                                                   ; A NODENAME OF 0-6 CHARACTERS
                      020F                                                   ;  IS ALLOWED
        06   53   D1  020F                  CMPL    R3,#6                    ; IS STRING TOO LONG?
             03   1B  0212                  BLEQU   10$                      ; NO, USE ENTERED VALUE
           FE23   31  0214                  BRW     PARSE_ERROR              ; YES
             03   BB  0217      10$:        PUSHR   #^M<R0,R1>               ; SAVE R0 AND R1
                      0219                                                   ; NODENAME IS STORED AS A
                      0219                                                   ; COUNTED ASCII STRING
 0000'CF     53   90  0219                  MOVB    R3,W^TST$GT_NODENAME     ; STORE LENGTH OF STRING
 0000'CF     53   28  021E                  MOVC3   R3,W^TST$GT_VALUE,-      ; STORE STRING
     0001'CF          0223                          W^TST$GT_NODENAME+1      ; NOTE R0 - R5 ARE DESTROYED!
             03   BA  0226                  POPR    #^M<R0,R1>               ; RESTORE R0 AND R1
                  05  0228                  RSB                              ; EXIT
                      0229
                      0229      NODISPLAY:                                   ; PROCESS NODISPLAY QUALIFIER
                      0229                  ASSUME  VAL_K_DISP_NO,EQ,0       ;
     0000'CF      94  0229                  CLRB    W^TST$GB_DISPLAY         ; UPDATE DISPLAY VALUE
                  05  022D                  RSB                              ; EXIT
                      022E
                      022E      NOFLOW:                                      ; PROCESS NOFLOW QUALIFIER
                      022E                  ASSUME  VAL_K_FLOW_NO,EQ,0       ;
     0000'CF      94  022E                  CLRB    W^TST$GB_FLOW            ; UPDATE FLOW CONTROL VALUE
                  05  0232                  RSB                              ; EXIT
                      0233
                      0233      NONAK:                                       ; PROCESS NONAK QUALIFIER
                      0233                  ASSUME  VAL_K_NAK_NO,EQ,0        ;
     0000'CF      94  0233                  CLRB    W^TST$GB_NAK             ; UPDATE NAK CONTROL VALUE
                  05  0237                  RSB                              ; EXIT
                      0238
                      0238      NOPRINT:                                     ; PROCESS NOPRINT QUALIFIER
                      0238                  ASSUME  VAL_K_PRIN_NO,EQ,0       ;
     0000'CF      94  0238                  CLRB    W^TST$GB_PRINT           ; UDPATE PRINT VALUE
                  05  023C                  RSB                              ; EXIT
                      023D
                      023D      NORETURN:                                    ; PROCESS NORETURN QUALIFIER
                      023D                  ASSUME  VAL_K_RETU_NO,EQ,0       ;
     0000'CF      94  023D                  CLRB    W^TST$GB_RETURN          ; UPDATE RETURN VALUE
```

```
                      05  0241   466            RSB                             ; EXIT
                          0242   467
                          0242   468 NOSTATISTICS:                              ; PROCESS NOSTATISTICS QUALIFIER
                          0242   469            ASSUME  VAL_K_STAT_NO,EQ,0      ;
         0000'CF          94  0242   470            CLRB    W^TST$GB_STAT           ; UPDATE STATISTICS VALUE
                      05  0246   471            RSB                             ; EXIT
                          0247   472
                          0247   473 PRINT:                                     ; PROCESS PRINT QUALIFIER
     0000'CF      80 8F   90  0247   474            MOVB    #VAL_K_PRIN_YES,W^TST$GB_PRINT ; UPDATE PRINT VALUE
                      05  024D   475            RSB                             ; EXIT
                          024E   476
                          024E   477 RETURN:                                    ; PROCESS RETURN QUALIFIER VALUE
          54   0000'CF    DE  024E   478            MOVAL   W^TST$AZ_RETURN,R4      ; GET ADDRESS OF KEYWORD TABLE
                    00DE  30  0253   479            BSBW    TST$MATCH               ; FIND TABLE INDEX OF KEYWORD
                      55  D6  0256   480            INCL    R5                      ;
         0000'CF      55  F6  0258   481            CVTLB   R5,W^TST$GB_RETURN      ; UPDATE RETURN USERDATA VALUE
                      05  025D   482            RSB                             ; EXIT
                          025E   483
                          025E   484 RQUEUE:                                    ; PROCESS RQUEUE QUALIFIER VALUE
              57      08  D0  025E   485            MOVL    #MAX_K_RQUEUE_DA,R7     ; DEFINE MAXIMUM VALUE FOR DATA TEST
              01      5A  91  0261   486            CMPB    R10,#VAL_K_TEST_DATA    ; IS IT A DATA TEST?
                      03  13  0264   487            BEQLU   10$                     ; BRANCH IF YES
              57      08  D0  0266   488            MOVL    #MAX_K_RQUEUE_IN,R7     ; NO, DEFINE MAX VALUE FOR INT TEST
                    00E5  30  0269   489 10$:       BSBW    TST$CVTU_DTB            ; CONVERT DIGITS TO BINARY VALUE
         0000'CF      56  F6  026C   490            CVTLB   R6,W^TST$GB_RQUEUE      ; UPDATE FLOW CONTROL VALUE
                      05  0271   491            RSB                             ; EXIT
                          0272   492
                          0272   493 SECONDS:                                   ; PROCESS SECONDS QUALIFIER VALUE
              7E      01  D0  0272   494            MOVL    #1,-(SP)                ; # SECONDS IN 1 SECOND
     57  00057E40 8F      D0  0275   495            MOVL    #MAX_K_TIME_DA,R7       ; DEFINE MAXIMUM SECOND VALUE
                          027C   496 TIME:                                      ; COMMON CODE
                    00D2  30  027C   497            BSBW    TST$CVTU_DTB            ; CONVERT DIGITS TO BINARY VALUE
   0000'CF      8E    56  C5  027F   498            MULL3   R6,(SP)+,W^TST$GL_SECONDS ; CALCULATE NUMBER OF SECONDS
                      05  0285   499            RSB                             ; EXIT
                          0286   500
                          0286   501 SIZE:                                      ; PROCESS SIZE QUALIFIER VALUE
          57    1000 8F   3C  0286   502            MOVZWL  #MAX_K_SIZE_DA,R7       ; DEFINE MAXIMUM VALUE FOR DATA TEST
              01      5A  91  028B   503            CMPB    R10,#VAL_K_TEST_DATA    ; IS IT A DATA TEST?
                      03  13  028E   504            BEQLU   10$                     ; BRANCH IF YES
              57      10  D0  0290   505            MOVL    #MAX_K_SIZE_IN,R7       ; NO, DEFINE MAX VALUE FOR INT TEST
                    00BB  30  0293   506 10$:       BSBW    TST$CVTU_DTB            ; CONVERT DIGITS TO BINARY VALUE
         0000'CF      56  F7  0296   507            CVTLW   R6,W^TST$GW_SIZE        ; UPDATE MESSAGE SIZE
                      05  029B   508            RSB                             ; EXIT
                          029C   509
                          029C   510 SPEED:                                     ; PROCESS SPEED QUALIFIER VALUE
     57  000F4240 8F      D0  029C   511            MOVL    #MAX_K_SPEED,R7         ; DEFINE MAXIMUM VALUE
                    00AB  30  02A3   512            BSBW    TST$CVTU_DTB            ; CONVERT DIGITS TO BINARY VALUE
         0000'CF      56  D0  02A6   513            MOVL    R6,W^TST$GL_SPEED       ; UPDATE BAUD RATE
                      05  02AB   514            RSB                             ; EXIT
                          02AC   515
                          02AC   516 SQUEUE:                                    ; PROCESS SQUEUE QUALIFIER
              57      08  D0  02AC   517            MOVL    #MAX_K_SQUEUE_DA,R7     ; DEFINE MAXIMUM VALUE FOR DATA TEST
              01      5A  91  02AF   518            CMPB    R10,#VAL_K_TEST_DATA    ; IS IT A DATA TEST?
                      03  13  02B2   519            BEQLU   10$                     ; BRANCH IF YES
              57      08  D0  02B4   520            MOVL    #MAX_K_SQUEUE_IN,R7     ; NO, DEFINE MAX VALUE FOR INT TEST
                    0097  30  02B7   521 10$:       BSBW    TST$CVTU_DTB            ; CONVERT DIGITS TO BINARY VALUE
         0000'CF      56  F6  02BA   522            CVTLB   R6,W^TST$GB_SQUEUE      ; UPDATE DTS QUEUE COUNT
```

```
                      05  02BF                RSB                                      ; EXIT
                          02C0
                          02C0    STATISTICS:                                          ; PROCESS STATISTICS QUALIFIER
 0000'CF      01      90  02C0                MOVB     #VAL_K_STAT_YES,W^TST$GB_STAT    ; UPDATE STATISTICS VALUE
                      05  02C5                RSB                                      ; EXIT
                          02C6
                          02C6    TYPE:                                                ; PROCESS TYPE QUALIFIER VALUE
                          02C6                $CASEB   SELECTOR=R10,DISPL=<-           ; TEST:
                          02C6                                   10$-                  ;  CONNECT TEST
                          02C6                                   20$-                  ;  DATA TEST
                          02C6                                   30$-                  ;  DISCONNECT TEST
                          02C6                                   40$-                  ;  INTERRUPT TEST
                          02C6                >                                        ;  MISCELLANEOUS TEST BELOW
 54      0000'CF      DE  02D2                MOVAL    W^TST$AZ_TYPE_MI,R4             ; GET ADDRESS OF KEYWORD TABLE
              1A      11  02D7                BRB      50$                             ; BRANCH TO COMMON CODE
 54      0000'CF      DE  02D9    10$:        MOVAL    W^TST$AZ_TYPE_CO,R4             ; GET ADDRESS OF KEYWORD TABLE
              13      11  02DE                BRB      50$                             ; BRANCH TO COMMON CODE
 54      0000'CF      DE  02E0    20$:        MOVAL    W^TST$AZ_TYPE_DA,R4             ; GET ADDRESS OF KEYWORD TABLE
              0C      11  02E5                BRB      50$                             ; BRANCH TO COMMON CODE
 54      0000'CF      DE  02E7    30$:        MOVAL    W^TST$AZ_TYPE_DI,R4             ; GET ADDRESS OF KEYWORD TABLE
              05      11  02EC                BRB      50$                             ; BRANCH TO COMMON CODE
 54      0000'CF      DE  02EE    40$:        MOVAL    W^TST$AZ_TYPE_IN,R4             ; GET ADDRESS OF KEYWORD TABLE
            003E      30  02F3    50$:        BSBW     TST$MATCH                       ; FIND TABLE INDEX OF KEYWORD
 0000'CF      55      F6  02F6                CVTLB    R5,W^TST$GB_TYPE                ; UPDATE MESSAGE TYPE
                      05  02FB                RSB                                      ; EXIT
            02FC          .SBTTL  TST$NEXTCHAR - EXAMINE NEXT CHARACTER
        000002FC          .PSECT  TST$CODE        NOWRT
            02FC
            02FC  ;++
            02FC  ; FUNCTIONAL DESCRIPTION:
            02FC  ;
            02FC  ;       TST$NEXTCHAR ATTEMPTS TO EXAMINE THE NEXT CHARACTER IN THE
            02FC  ;       BUFFER. IF THE END OF THE BUFFER HAS BEEN REACHED, TST$NEXTCHAR
            02FC  ;       SIGNALS END OF LINE CONDITION; OTHERWISE THE NEXT CHARACTER
            02FC  ;       FOUND IS RETURNED ALONG WITH A VALUE INDICATING WHAT TYPE OF
            02FC  ;       CHARACTER IT IS.
            02FC  ;
            02FC  ; CALLING SEQUENCE:
            02FC  ;
            02FC  ;       BSB/JSB TST$NEXTCHAR
            02FC  ;
            02FC  ; INPUT PARAMETERS:
            02FC  ;
            02FC  ;       R8      THE ADDRESS OF THE NEXT CHARACTER IN THE BUFFER
            02FC  ;       R9      THE ADDRESS OF THE END OF THE BUFFER + 1
            02FC  ;
            02FC  ; IMPLICIT INPUTS:
            02FC  ;
            02FC  ;       NONE
            02FC  ;
            02FC  ; OUTPUT PARAMETERS:
            02FC  ;
            02FC  ;       R0      RESULT WHERE:
            02FC  ;               0 = END OF LINE OR CHARACTER IS AN EXCLAMATION OR DASH
            02FC  ;               1 = CHARACTER IS A SLASH
            02FC  ;               2 = CHARACTER IS AN EQUAL SIGN OR COLON
            02FC  ;               3 = CHARACTER IS A SPACE OR TAB
            02FC  ;               4 = CHARACTER IS NONE OF THE ABOVE
            02FC  ;       R1      THE CHARACTER EXAMINED (0 OR "NULL" IF END OF LINE)
            02FC  ;       R8      UPDATED NEXT CHARACTER POINTER
            02FC  ;
            02FC  ; IMPLICIT OUTPUTS:
            02FC  ;
            02FC  ;       NONE
            02FC  ;
            02FC  ; COMPLETION CODES:
            02FC  ;
            02FC  ;       NONE
            02FC  ;
            02FC  ; SIDE EFFECTS:
            02FC  ;
            02FC  ;       NONE
            02FC  ;
            02FC  ;--
            02FC
            02FC  TST$NEXTCHAR::                          ; CONTROL POINT
    50  D4  02FC          CLRL    R0                      ; INITIALIZE RETURN VALUE
    51  D4  02FE          CLRL    R1                      ; SET R1 TO "NULL"
            0300  ; ***** R0 = 0
59  58  D1  0300          CMPL    R8,R9                   ; END OF COMMAND LINE?
    2E  13  0303          BEQLU   10$                     ; YES
51  88  9A  0305          MOVZBL  (R8)+,R1                ; GET NEXT CHARACTER
```

```
         21   51   91  0308    606           CMPB    R1,#^A\!\                     ; IS IT AN EXCLAMATION POINT?
              26   13  030B    607           BEQLU   10$                           ; YES, IGNORE REST OF LINE
         2D   51   91  030D    608           CMPB    R1,#^A\-\                     ; IS IT A DASH?
              21   13  0310    609           BEQLU   10$                           ; YES, IGNORE REST OF LINE
                       0312    610 ; ***** R0 = 1
              50   D6  0312    611           INCL    R0                            ; INCREMENT RETURN VALUE
         2F   51   91  0314    612           CMPB    R1,#^A\/\                     ; IS IT A SLASH?
              1A   13  0317    613           BEQLU   10$                           ; YES
                       0319    614 ; ***** R0 = 2
              50   D6  0319    615           INCL    R0                            ; INCREMENT RETURN VALUE
         3D   51   91  031B    616           CMPB    R1,#^A\=\                     ; IS IT AN EQUALS_SIGN?
              13   13  031E    617           BEQLU   10$                           ; YES
         3A   51   91  0320    618           CMPB    R1,#^A\:\                     ; IS IT A COLON?
              0E   13  0323    619           BEQLU   10$                           ; YES
                       0325    620 ; ***** R0 = 3
              50   D6  0325    621           INCL    R0                            ; INCREMENT RETURN VALUE
         20   51   91  0327    622           CMPB    R1,#^A\ \                     ; IS IT A SPACE?
              07   13  032A    623           BEQLU   10$                           ; YES
         09   51   91  032C    624           CMPB    R1,#^X09                      ; IS IT A TAB?
              02   13  032F    625           BEQLU   10$                           ; YES
                       0331    626 ; ***** R0 = 4
              50   D6  0331    627           INCL    R0                            ; IT'S NONE OF THE ABOVE
                   05  0333    628 10$:      RSB                                   ; EXIT
```

```
                          0334          .SBTTL  TST$MATCH - KEYWORD MATCH ROUTINE
                      00000334          .PSECT  TST$CODE        NOWRT
                          0334
                          0334  ;++
                          0334  ; FUNCTIONAL DESCRIPTION:
                          0334  ;
                          0334  ;       TST$MATCH SEARCHES THE SPECIFIED KEYWORD TABLE FOR A MATCH WITH
                          0334  ;       THE SPECIFIED KEYWORD STRING. IF A MATCH IS FOUND, THE INDEX
                          0334  ;       OF THE MATCHING TABLE ENTRY IS RETURNED; OTHERWISE CONTROL IS
                          0334  ;       TRANSFERRED TO AN ERROR ROUTINE. THE END OF THE TABLE IS MARKED
                          0334  ;       BY A NULL KEYWORD ENTRY, I.E., A NULL COUNTED ASCII STRING.
                          0334  ;
                          0334  ; CALLING SEQUENCE:
                          0334  ;
                          0334  ;       BSB/JSB TST$MATCH
                          0334  ;
                          0334  ; INPUT PAPAMETERS:
                          0334  ;
                          0334  ;       R4      ADDRESS OF THE KEYWORD TABLE TO SEARCH
                          0334  ;       R6      ADDRESS OF THE KEYWORD STRING TO MATCH AGAINST THE TABLE
                          0334  ;
                          0334  ; IMPLICIT INPUTS:
                          0334  ;
                          0334  ;       NONE
                          0334  ;
                          0334  ; OUTPUT PARAMETERS:
                          0334  ;
                          0334  ;       R5      THE INDEX OF THE MATCHING TABLE ENTRY IF A MATCH IS FOUND;
                          0334  ;               OTHERWISE UNDEFINED.
                          0334  ;
                          0334  ; IMPLICIT OUTPUTS:
                          0334  ;
                          0334  ;       NONE
                          0334  ;
                          0334  ; COMPLETION CODES:
                          0334  ;
                          0334  ;       NONE
                          0334  ;
                          0334  ; SIDE EFFECTS:
                          0334  ;
                          0334  ;       CONTROL IS TRANSFERRED TO PARSE_ERROR IF AN ERROR IS DETECTED.
                          0334  ;
                          0334  ;--
                          0334
                          0334  TST$MATCH::                                     ; CONTROL POINT
              1F    BB    0334          PUSHR   #^M<R0,R1,R2,R3,R4>     ; SAVE REGISTERS
              55    D4    0336          CLRL    R5                      ; INITIALIZE RETURN VALUE
        50    84    90    0338  10$:    MOVB    (R4)+,R0                ; GET # OF SIGNIFICANT CHARACTERS
              11    13    033B          BEQLU   30$                     ; END OF TABLE IF ZERO
  66    64    50    29    033D          CMPC3   R0,(R4),(R6)            ; DO STRINGS MATCH?
              08    13    0341          BEQLU   20$                     ; YES
  54    51    50    C1    0343          ADDL3   R0,R1,R4                ; SET POINTER TO NEXT ENTRY
              55    D6    0347          INCL    R5                      ; INCREMENT TABLE INDEX
              ED    11    0349          BRB     10$                     ; TRY AGAIN
              1F    BA    034B  20$:    POPR    #^M<R0,R1,R2,R3,R4>     ; RESTORE REGISTERS
                    05    034D          RSB                             ; EXIT
            FCE9    31    034E  30$:    BRW     PARSE_ERROR             ; BRANCH TO ERROR ROUTINE
```

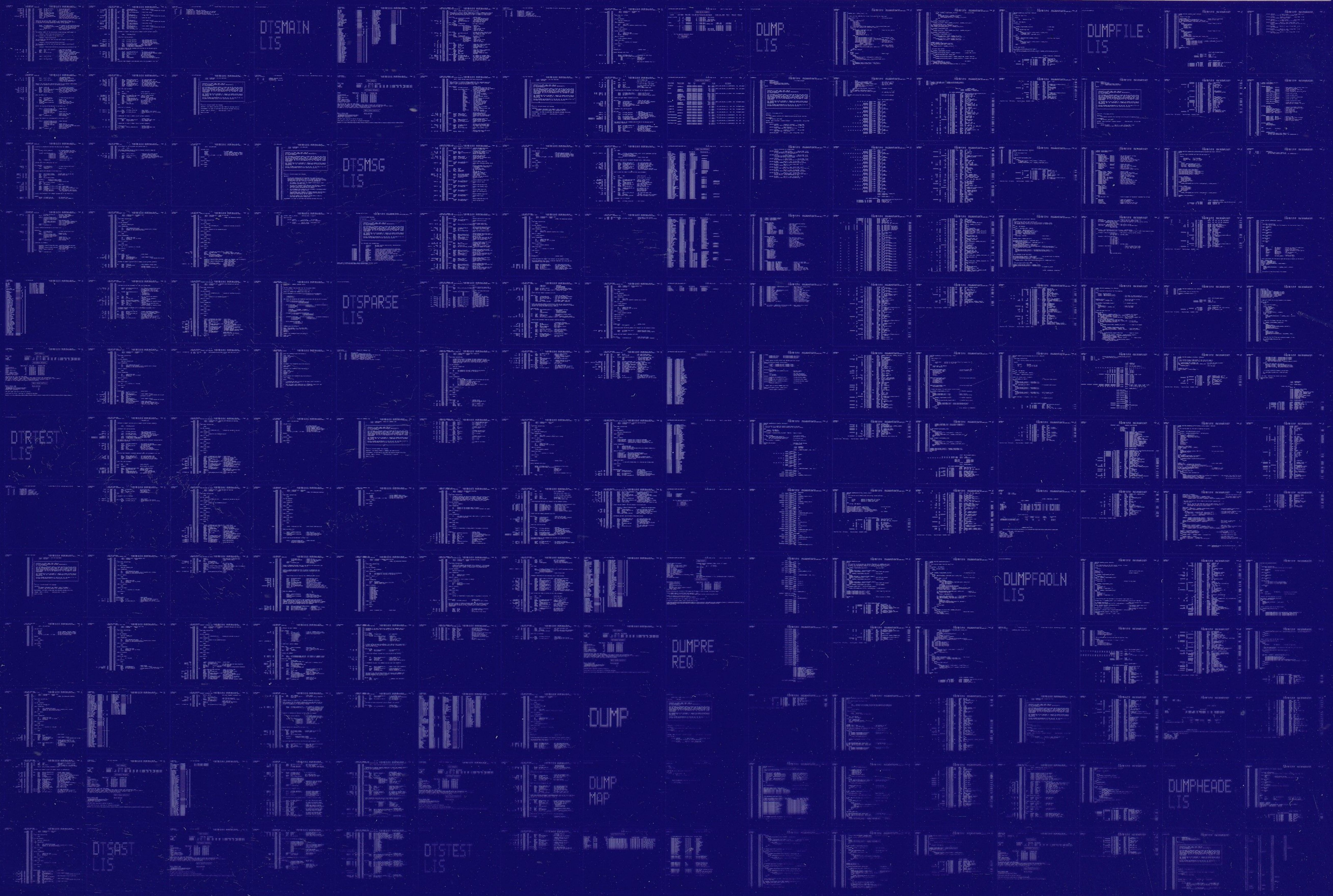
```
                             0351          .SBTTL  TST$CVTU_DTB - CONVERT UNSIGNED DECIMAL TO BINARY
                         00000351          .PSECT  TST$CODE        NOWRT
                             0351
                             0351   ;++
                             0351   ; FUNCTIONAL DESCRIPTION:
                             0351   ;
                             0351   ;       TST$CVTU_DTB CONVERTS AN UNSIGNED ASCII STRING OF 1 TO 8 DECIMAL
                             0351   ;       DIGITS TO A 32-BIT BINARY VALUE. IF THE RESULTANT VALUE EXCEEDS
                             0351   ;       THE GIVEN LIMIT, CONTROL IS TRANSFERRED TO AN ERROR ROUTINE.
                             0351   ;
                             0351   ; CALLING SEQUENCE:
                             0351   ;
                             0351   ;       BSB/JSB TST$CVTU_DTB
                             0351   ;
                             0351   ; INPUT PARAMETERS:
                             0351   ;
                             0351   ;       R3      LENGTH OF ASCII STRING
                             0351   ;       R7      MAXIMUM VALUE
                             0351   ;
                             0351   ; IMPLICIT INPUTS:
                             0351   ;
                             0351   ;       TST$GT_VALUE = ASCII STRING TO CONVERT
                             0351   ;
                             0351   ; OUTPUT PARAMETERS:
                             0351   ;
                             0351   ;       R2-R5   DESTROYED
                             0351   ;       R6      BINARY VALUE OF STRING
                             0351   ;       R7      UNCHANGED
                             0351   ;
                             0351   ; IMPLICIT OUTPUTS:
                             0351   ;
                             0351   ;       NONE
                             0351   ;
                             0351   ; COMPLETION CODES:
                             0351   ;
                             0351   ;       NONE
                             0351   ;
                             0351   ; SIDE EFFECTS:
                             0351   ;
                             0351   ;       CONTROL IS TRANSFERRED TO PARSE_ERROR IF AN ERROR IS DETECTED.
                             0351   ;
                             0351   ;--
                             0351
                             0351   TST$CVTU_DTB::                                  ; CONTROL POINT
     52   0000'CF     DE     0351           MOVAL   W^TST$GT_VALUE,R2               ; GET ADDRESS OF ASCII STRING
                             0356
                             0356   ;
                             0356   ; THE ASCII STRING IS STORED IN REVERSE ORDER, SO THE POINTER IS PLACED
                             0356   ; ONE PAST THE END OF THE STRING. THEREFORE, THE STRING IS SCANNED IN
                             0356   ; REVERSE ORDER TO OBTAIN THE LEAST-SIGNIFICANT TO MOST-SIGNIFICANT
                             0356   ; CHARACTERS.
                             0356   ;
                             0356
             52    53     C0     0356           ADDL2   R3,R2                           ; ADD STRING LENGTH TO POINTER
             08    53     91     0359           CMPB    R3,#8                           ; IS STRING TOO LONG?
                   22     1A     035C           BGTRU   20$                             ; YES
             54    01     9A     035E           MOVZBL  #1,R4                           ; SET-UP DIGIT PLACE VALUE
             56  D4  0361     745          CLRL    R6                      ; ZERO RETURN VALUE
        55   72  9A  0363     746 10$:     MOVZBL  -(R2),R5                ; GET NEXT ASCII CHARACTER
        55   30  82  0366     747          SUBB2   #^X30,R5                ; CONVERT ASCII DIGIT TO BINARY
        09   55  91  0369     748          CMPB    R5,#9                   ; IS IT NON-NUMERIC?
             12  1A  036C     749          BGTRU   20$                     ; YES
        55   54  C4  036E     750          MULL2   R4,R5                   ; MULTIPLY DIGIT BY ITS PLACE VALUE
        56   55  C0  0371     751          ADDL2   R5,R6                   ; ADD THIS TO THE TOTAL
        54   0A  C4  0374     752          MULL2   #10,R4                  ; MULTIPLY PLACE VALUE BY 10
          E9 53  F5  0377     753          SOBGTR  R3,10$                  ; ANOTHER DIGIT TO CONVERT?
        57   56  D1  037A     754          CMPL    R6,R7                   ; IS CONVERTED VALUE TOO LARGE?
             01  1A  037D     755          BGTRU   20$                     ; YES
                 05  037F     756          RSB                             ; NO, EXIT
           FCB7  31  0380     757 20$:     BRW     PARSE_ERROR             ; BRANCH TO ERROR ROUTINE
                     0383     758          .END
```

```
$$COUNT        = 00000004               P                00000000 RG    02      TST$PARSE        00000000 RG    02
BACK             000001BD R     02      PARAM            00000021 R     02      TYPE             000002C6 R     02
CONNTEST         0000008E R     02      PARAMETER        0000003E R     02      VALUE_CHAR       00000155 R     02
DATATEST         000000A2 R     02      PARAM_CHAR       0000005F R     02      VALUE_LOOP       00000144 R     02
DFT_K_BACK     = 00000000               PARAM_DELIMITER  0000006B R     02      VAL_K_BACK_NO  = 00000000
DFT_K_FLOW     = 00000002               PARAM_LOOP       0000004E R     02      VAL_K_DISP_NO  = 00000000
DFT_K_NAK      = 00000000               PARSE_ERROR      0000003A R     02      VAL_K_FLOW_MESS= 00000002
DFT_K_RETURN_CO= 00000000               PRINT            00000247 R     02      VAL_K_FLOW_NO  = 00000000
DFT_K_RETURN_DI= 00000000               QUAL             00000018 R     02      VAL_K_NAK_NO   = 00000000
DFT_K_RQUEUE_DA= 00000001               QUALIFIER        0000011C R     02      VAL_K_PRIN_NO  = 00000000
DFT_K_RQUEUE_IN= 00000001               QUAL_CHAR        00000161 R     02      VAL_K_PRIN_YES = 00000080
DFT_K_SIZE_DA  = 00000080               QUAL_DELIMITER   0000016D R     02      VAL_K_RETU_NO  = 00000000
DFT_K_SIZE_IN  = 00000010               QUAL_DISPATCH    00000186 R     02      VAL_K_STAT_NO  = 00000000
DFT_K_SQUEUE_DA= 00000001               QUAL_LOOP        0000012C R     02      VAL_K_STAT_YES = 00000001
DFT_K_SQUEUE_IN= 00000001               QUAL_REEXAMINE   0000012F R     02      VAL_K_TEST_DATA= 00000001
DFT_K_TIME_DA  = 0000001E               QUAL_VALUE       0000013D R     02      VAL_K_TYPE_ABRT= 00000001
DFT_K_TIME_IN  = 0000001E               REEXAMINE_CHAR   00000005 R     02      VAL_K_TYPE_ACCE= 00000001
DFT_K_TYPE_CO  = 00000001               RETURN           0000024E R     02      VAL_K_TYPE_NAME= 00000000
DFT_K_TYPE_DA  = 00000000               RQUEUE           0000025E R     02      VAL_K_TYPE_SINK= 00000000
DFT_K_TYPE_DI  = 00000001               SECONDS          00000272 R     02      VLD_M_BACK     = 00000001
DFT_K_TYPE_IN  = 00000000               SIZ...         = 00000001               VLD_M_FLOW     = 00000004
DFT_K_TYPE_MI  = 00000000               SIZE             00000286 R     02      VLD_M_HOURS    = 00000008
DISCTEST         000000D6 R     02      SPACE_OR_TAB     00000013 R     02      VLD_M_MINUTES  = 00000010
DISPLAY          000001CA R     02      SPEED            0000029C R     02      VLD_M_NAK      = 00000020
END_OF_LINE      0000002C R     02      SQUEUE           000002AC R     02      VLD_M_NOBACK   = 00000040
FLG_M_DELIMITER= 00000008               STATISTICS       000002C0 R     02      VLD_M_NOFLOW   = 00000200
FLG_M_MULTILINE= 00000002               TIME             0000027C R     02      VLD_M_NONAK    = 00000400
FLG_M_PARSERROR= 00000001               TST$AZ_FLOW      ********   X   02      VLD_M_NORETURN = 00001000
FLG_V_DELIMITER= 00000003               TST$AZ_PARAM     ********   X   02      VLD_M_RETURN   = 00008000
FLG_V_PARAMETER= 00000002               TST$AZ_QUAL      ********   X   02      VLD_M_RQUEUE   = 00010000
FLOW             000001D6 R     02      TST$AZ_RETURN    ********   X   02      VLD_M_SECONDS  = 00020000
HOURS            000001E6 R     02      TST$AZ_TYPE_CO   ********   X   02      VLD_M_SIZE     = 00040000
INTETEST         000000EA R     02      TST$AZ_TYPE_DA   ********   X   02      VLD_M_SQUEUE   = 00100000
K_LIST_MEB     = 00000000               TST$AZ_TYPE_DI   ********   X   02      VLD_M_TYPE     = 00400000
MAX_K_BACK     = 00000080               TST$AZ_TYPE_IN   ********   X   02
MAX_K_DISPLAY  = 00000026               TST$AZ_TYPE_MI   ********   X   02
MAX_K_NAK      = 00000080               TST$CVTU_DTB     00000351 RG    02
MAX_K_RQUEUE_DA= 00000008               TST$GB_BACK      ********   X   02
MAX_K_RQUEUE_IN= 00000008               TST$GB_DISPLAY   ********   X   02
MAX_K_SIZE_DA  = 00001000               TST$GB_FLOW      ********   X   02
MAX_K_SIZE_IN  = 00000010               TST$GB_NAK       ********   X   02
MAX_K_SPEED    = 000F4240               TST$GB_PRINT     ********   X   02
MAX_K_SQUEUE_DA= 00000008               TST$GB_RETURN    ********   X   02
MAX_K_SQUEUE_IN= 00000008               TST$GB_RQUEUE    ********   X   02
MAX_K_TIME_DA  = 00057E40               TST$GB_SQUEUE    ********   X   02
MINUTES          000001F2 R     02      TST$GB_STAT      ********   X   02
MISCTEST         0000010D R     02      TST$GB_TEST      ********   X   02
NAK              000001FD R     02      TST$GB_TYPE      ********   X   02
NEXT_ELEMENT     00000002 R     02      TST$GL_SECONDS   ********   X   02
NOBACK           0000020A R     02      TST$GL_SPEED     ********   X   02
NODENAME         0000020F R     02      TST$GL_VALID     ********   X   02
NODISPLAY        00000229 R     02      TST$GT_KEYWORD   ********   X   02
NOFLOW           0000022E R     02      TST$GT_NODENAME  ********   X   02
NONAK            00000233 R     02      TST$GT_VALUE     ********   X   02
NOPRINT          00000238 R     02      TST$GW_SIZE      ********   X   02
NORETURN         0000023D R     02      TST$MATCH        00000334 RG    02
NOSTATISTICS     00000242 R     02      TST$NEXTCHAR     000002FC RG    02
```

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

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           00000000  (     0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
TST$CODE                        00000383  (   899.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC BYTE

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   34    00:00:00.05    00:00:01.79
Command processing              118    00:00:00.60    00:00:04.84
Pass 1                          215    00:00:06.14    00:00:16.75
Symbol table sort                 0    00:00:00.24    00:00:00.26
Pass 2                          145    00:00:02.07    00:00:05.58
Symbol table output              13    00:00:00.09    00:00:00.11
Psect synopsis output             2    00:00:00.03    00:00:00.03
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            529    00:00:09.23    00:00:29.38

The working set limit was 1350 pages.
29598 bytes (58 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 200 non-local and 24 local symbols.
820 source lines were read in Pass 1, producing 21 object records in Pass 2.
23 pages of virtual memory were used to define 19 macros.

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

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[DTSDTR.OBJ]DTSDTR.MLB;1                  6
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     6
TOTALS (all libraries)                               12

223 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:DTSPARSE/OBJ=OBJ$:DTSPARSE MSRC$:DTPREFIX/UPDATE=(ENH$:DTPREFIX)+MSRC$:DTSPARSE/UPDATE=(ENH$:DTSPARSE)
```

0123 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



DTSMAIN
LIS

DUMP
LIS

DUMPFILE
LIS

DTSMSG
LIS

DTSPARSE
LIS

DTRTEST
LIS

DUMPFAOLN
LIS

DUMPRE
REQ

DUMP

DUMP
MAP

DUMPHEADE
LIS

DTSAST
LIS

DTSTEST
LIS