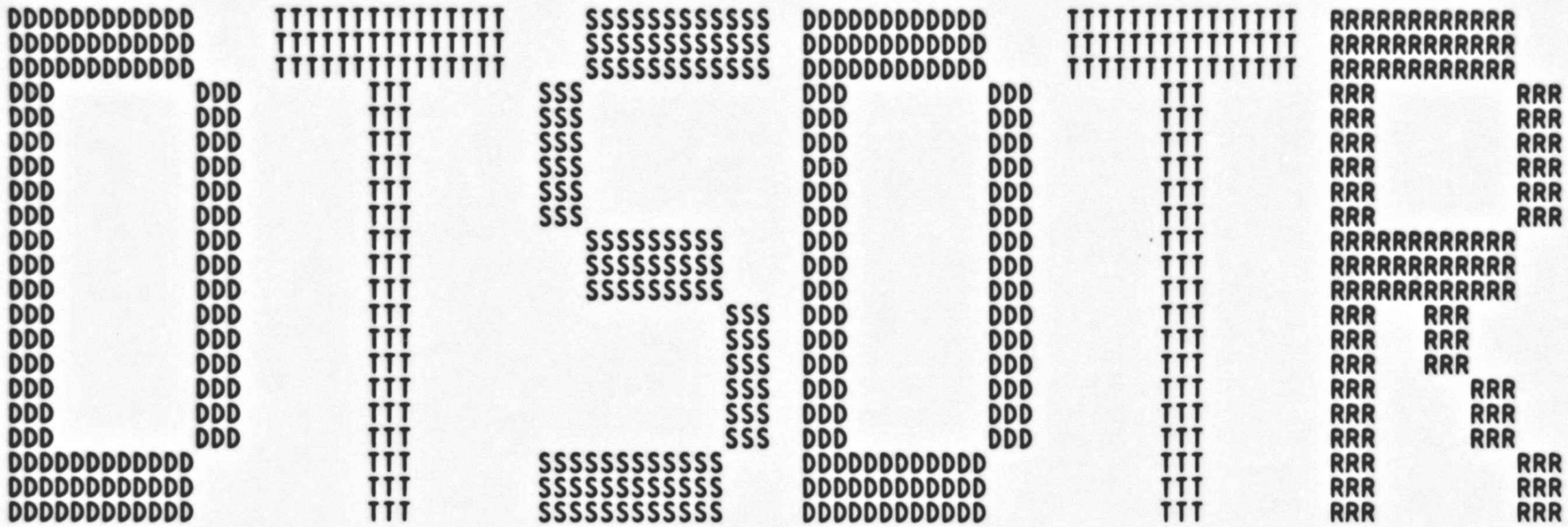
DDDDDDDDDDDD TTTTTTTTTTTTTTT SSSSSSSSSSS DDDDDDDDDDDD TTTTTTTTTTTTTTT RRRRRRRRRRR
DDDDDDDDDDDD TTTTTTTTTTTTTTT SSSSSSSSSSS DDDDDDDDDDDD TTTTTTTTTTTTTTT RRRRRRRRRRR
DDDDDDDDDDDD TTTTTTTTTTTTTTT SSSSSSSSSSS DDDDDDDDDDDD TTTTTTTTTTTTTTT RRRRRRRRRRR
DDD DDD TTT SSS DDD DDD TTT RRR RRR
DDD DDD TTT SSS DDD 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 RRRRRRRRRRR
DDD DDD TTT SSSSSSSSS DDD DDD TTT RRRRRRRRRRR
DDD DDD TTT SSSSSSSSS DDD DDD TTT RRRRRRRRRRR
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 SSSSSSSSSSS DDDDDDDDDDDD TTT RRR RRR
DDDDDDDDDDDD TTT SSSSSSSSSSS DDDDDDDDDDDD TTT RRR RRR
DDDDDDDDDDDD TTT SSSSSSSSSSS DDDDDDDDDDDD TTT RRR RRR

FILEID**DTCOMMON

```
DDDDDDDD    TTTTTTTTTT    CCCCCCCC    OOOOOO    MM      MM  MM      MM    OOOOOO    NN      NN
DDDDDDDD    TTTTTTTTTT    CCCCCCCC    OOOOOO    MM      MM  MM      MM    OOOOOO    NN      NN
DD      DD      TT      CC          OO      OO  MMMM  MMMM  MMMM  MMMM  OO      OO  NN      NN
DD      DD      TT      CC          OO      OO  MMMM  MMMM  MMMM  MMMM  OO      OO  NN      NN
DD      DD      TT      CC          OO      OO  MM  MM  MM  MM  MM  MM  OO      OO  NNNN    NN
DD      DD      TT      CC          OO      OO  MM  MM  MM  MM  MM  MM  OO      OO  NNNN    NN
DD      DD      TT      CC          OO      OO  MM      MM  MM      MM  OO      OO  NN  NN  NN
DD      DD      TT      CC          OO      OO  MM      MM  MM      MM  OO      OO  NN  NN  NN
DD      DD      TT      CC          OO      OO  MM      MM  MM      MM  OO      OO  NN    NNNN
DD      DD      TT      CC          OO      OO  MM      MM  MM      MM  OO      OO  NN    NNNN
DD      DD      TT      CC          OO      OO  MM      MM  MM      MM  OO      OO  NN      NN
DD      DD      TT      CC          OO      OO  MM      MM  MM      MM  OO      OO  NN      NN
DDDDDDDD        TT        CCCCCCCC    OOOOOO    MM      MM  MM      MM    OOOOOO    NN      NN
DDDDDDDD        TT        CCCCCCCC    OOOOOO    MM      MM  MM      MM    OOOOOO    NN      NN

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

Table of contents

```
0000     1          .TITLE  TST$DTCOMMON - COMMON ROUTINES FOR DTS/DTR
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28
0000    29  ;++
0000    30  ; FACILITY: DTS/DTR DECNET TEST PACKAGE
0000    31  ;
0000    32  ; ABSTRACT: MISCELLANEOUS ROUTINES COMMON TO DTS/DTR.
0000    33  ;
0000    34  ; ENVIRONMENT: DTS/DTR RUN IN USER MODE AND REQUIRE NETWORK PRIVILEGE.
0000    35  ;
0000    36  ; AUTHOR: JAMES A. KRYCKA,      CREATION DATE: 11-AUG-77
0000    37  ;
0000    38  ; MODIFICATIONS:
0000    39  ;
0000    40  ;       V02-003 SGD2003         Scott G. Davis  17-Nov-1980
0000    41  ;               Add check for new code - SS$_LINKABORT
0000    42  ;       V02-002 SGD2002         Scott G. Davis  29-Sep-1980
0000    43  ;               Get around problem with multiple outstanding I/O
0000    44  ;--
```

```
0000     46          .SBTTL  DECLARATIONS
0000     47
0000     48 ;
0000     49 ; INCLUDE FILES:
0000     50 ;
0000     51          EFNDEF                          ; DEFINE EFN'S AND FUNCTION CODES
0000     52          $QIODEF                         ; DEFINE QIO OFFSETS
0000     53          $RABDEF                         ; DEFINE RAB OFFSETS
0000     54          $RMSDEF                         ; DEFINE RMS COMPLETION CODES
0000     55          $SSDEF                          ; DEFINE SYSTEM SERVICE STATUS CODES
0000     56          .IIF NE K_LIST_MEB, .LIST MEB   ; DEFINED IN DTPREFIX.MAR
0000     57 ;
0000     58 ; MACROS:
0000     59 ;
0000     60 ;        NONE
0000     61 ;
0000     62 ; EQUATED SYMBOLS:
0000     63 ;
0000     64 ;        NONE
0000     65 ;
0000     66 ; OWN STORAGE:
0000     67 ;
0000     68 ;        NONE
```

```
                     70          .SBTTL  TST$CHECK_SS - CHECK SYSTEM SERVICE STATUS CODE
00000000             71          .PSECT  TST$CODE        NOWRT
                     72 C::                                      ; SYMBOL FOR DEBUGGING PURPOSES
                     73
                     74 ;++
                     75 ; FUNCTIONAL DESCRIPTION:
                     76 ;
                     77 ;       TST$CHECK_SS CHECKS THE STATUS CODE IN R0 FOLLOWING A CALL TO A
                     78 ;       SYSTEM SERVICE. IF FAILURE (EXCEPT AS NOTED BELOW) IS INDICATED
                     79 ;       THE IMAGE IS TERMINATED WITH R0 AS THE EXIT COMPLETION CODE.
                     80 ;
                     81 ; CALLING SEQUENCE:
                     82 ;
                     83 ;       BSB/JSB TST$CHECK_SS
                     84 ;
                     85 ; INPUT PARAMETERS:
                     86 ;
                     87 ;       R0      SYSTEM SERVICE STATUS CODE
                     88 ;
                     89 ; IMPLICIT INPUTS:
                     90 ;
                     91 ;       NONE
                     92 ;
                     93 ; OUTPUT PARAMETERS:
                     94 ;
                     95 ;       R1      TST$CHECK_SS COMPLETION CODE
                     96 ;
                     97 ; IMPLICIT OUTPUTS:
                     98 ;
                     99 ;       NONE
                    100 ;
                    101 ; COMPLETION CODES:
                    102 ;
                    103 ;       R1      0 = STATUS CODE IS ABORT (SS$_ABORT) OR
                    104 ;                   STATUS CODE IS CANCEL (SS$_CANCEL) OR
                    105 ;                   STATUS CODE IS REJECT (SS$_REJECT) OR
                    106 ;                   STATUS CODE IS FILE NOT ACCESSED (SS$_FILNOTACC)
                    107 ;               1 = SUCCESS
                    108 ;
                    109 ; SIDE EFFECTS:
                    110 ;
                    111 ;       IF THE STATUS CODE INDICATES FAILURE (EXCEPT AS NOTED ABOVE),
                    112 ;       THE IMAGE IS TERMINATED WITH THE STATUS CODE AS THE EXIT
                    113 ;       COMPLETION CODE.
                    114 ;
                    115 ;--
                    116
                    117 TST$CHECK_SS::                               ; CONTROL POINT
     51    01    D0 0000    118          MOVL    #1,R1               ; SET RETURN CODE TO SUCCESS
20E4 8F    50    B1 0003    119          CMPW    R0,#<SS$_LINKABORT&^XFFFF> ; No, Check for aborted I/O
     1F    13    13 0008    120          BEQLU   10$                 ; If EQL nonfatal
     2C    50    B1 000A    121          CMPW    R0,#<SS$_ABORT&^XFFFF>  ; NO, CHECK FOR ABORTED I/O
     1A    13    13 000D    122          BEQLU   10$                 ; NON-FATAL IF ABORTED
0830 8F    50    B1 000F    123          CMPW    R0,#<SS$_CANCEL&^XFFFF> ; NO, CHECK FOR CANCELLED I/O
     13    13    13 0014    124          BEQLU   10$                 ; NON-FATAL IF CANCELLED
0294 8F    50    B1 0016    125          CMPW    R0,#<SS$_REJECT&^XFFFF> ; NO, CHECK FOR CONNECT REJECTED
     0C    13    13 001B    126          BEQLU   10$                 ; NON-FATAL IF CONNECT REJECTED
```

```
                00AC 8F    50    B1  001D   127          CMPW    R0,#<SS$_FILNOTACC&^XFFFF> ; NO, CHECK FOR FILE NOT ACCESSED
                           05    13  0022   128          BEQLU   10$                        ; OCCURS IF DTR HAS EXITED
                     50    01    D0  0024   129          MOVL    S^#SS$_NORMAL,R0           ; Treat as success
                           02    11  0027   130          BRB     20$                        ; Take a common exit
                           51    D4  0029   131 10$:      CLRL    R1                         ; SET RETURN CODE TO FAILURE
                           05    002B   132 20$:      RSB                                ; EXIT
```

```
                     002C   134          .SBTTL  TST$CHECK_RMS - CHECK RMS COMPLETION CODE
                0000002C    135          .PSECT  TST$CODE        NOWRT
                     002C   136
                     002C   137 ;++
                     002C   138 ; FUNCTIONAL DESCRIPTION:
                     002C   139 ;
                     002C   140 ;       TST$CHECK_RMS CHECKS THE COMPLETION CODE IN R0 FOLLOWING A CALL
                     002C   141 ;       TO RMS. IF FAILURE (EXCEPT AS NOTED BELOW) IS INDICATED
                     002C   142 ;       THE IMAGE IS TERMINATED WITH R0 AS THE EXIT COMPLETION CODE.
                     002C   143 ;
                     002C   144 ; CALLING SEQUENCE:
                     002C   145 ;
                     002C   146 ;       BSB/JSB TST$CHECK_RMS
                     002C   147 ;
                     002C   148 ; INPUT PARAMETERS:
                     002C   149 ;
                     002C   150 ;       R0      RMS COMPLETION CODE
                     002C   151 ;
                     002C   152 ; IMPLICIT INPUTS:
                     002C   153 ;
                     002C   154 ;       NONE
                     002C   155 ;
                     002C   156 ; OUTPUT PARAMETERS:
                     002C   157 ;
                     002C   158 ;       R1      TST$CHECK_RMS COMPLETION CODE
                     002C   159 ;
                     002C   160 ; IMPLICIT OUTPUTS:
                     002C   161 ;
                     002C   162 ;       NONE
                     002C   163 ;
                     002C   164 ; COMPLETION CODES:
                     002C   165 ;
                     002C   166 ;       R1      0 = RMS COMPLETION CODE IS END-OF-FILE (RMS$_EOF) OR
                     002C   167 ;                   RMS COMPLETION CODE IS TIME-OUT (RMS$_TMO)
                     002C   168 ;               1 = SUCCESS
                     002C   169 ;
                     002C   170 ; SIDE EFFECTS:
                     002C   171 ;
                     002C   172 ;       IF THE RMS COMPLETION CODE INDICATES FAILURE (EXCPET AS NOTED
                     002C   173 ;       ABOVE) THE IMAGE IS TERMINATED WITH R0 AS THE EXIT COMPLETION CODE.
                     002C   174 ;
                     002C   175 ;--
                     002C   176
                     002C   177 TST$CHECK_RMS::                                  ; CONTROL POINT
          51   01    D0     002C   178          MOVL    #1,R1                    ; SET RETURN CODE TO SUCCESS
            19 50    E8     002F   179          BLBS    R0,20$                   ; WAS RMS FUNCTION SUCCESSFUL?
827A 8F     50    B1     0032   180          CMPW    R0,#<RMS$_EOF&^XFFFF>    ; NO, CHECK FOR END-OF-FILE
               10    13     0037   181          BEQLU   10$                      ; NON-FATAL IF END-OF-FILE
81B0 8F     50    B1     0039   182          CMPW    R0,#<RMS$_TMO&^XFFFF>    ; NO, CHECK FOR TIME-OUT
               09    13     003E   183          BEQLU   10$                      ; NON-FATAL IF TIME-OUT
                     0040   184          $EXIT_S R0                       ; TERMINATE THE IMAGE!!
               51    D4     0049   185 10$:     CLRL    R1                       ; SET RETURN CODE TO FAILURE
                     05     004B   186 20$:     RSB                              ; EXIT
```

```
                        004C   188          .SBTTL  TST$CHECK_IOSB - CHECK I/O STATUS BLOCK CODE
                    0000004C   189          .PSECT  TST$CODE        NOWRT
                        004C   190
                        004C   191 ;++
                        004C   192 ; FUNCTIONAL DESCRIPTION:
                        004C   193 ;
                        004C   194 ;       TST$CHECK_IOSB CHECKS THE STATUS CODE IN THE SPECIFIED I/O STATUS
                        004C   195 ;       BLOCK FOLLOWING A CALL TO THE QIO SYSTEM SERVICE. IF FAILURE
                        004C   196 ;       (EXCPET AS NOTED BELOW) IS INDICATED, THE IMAGE IS TERMINATED
                        004C   197 ;       WITH THE I/O STATUS CODE AS THE EXIT COMPLETION CODE.
                        004C   198 ;
                        004C   199 ; CALLING SEQUENCE:
                        004C   200 ;
                        004C   201 ;       BSB/JSB TST$CHECK_IOSB
                        004C   202 ;
                        004C   203 ; INPUT PARAMETERS:
                        004C   204 ;
                        004C   205 ;       R0      ADDRESS OF IOSB TO EXAMINE
                        004C   206 ;
                        004C   207 ; IMPLICIT INPUTS:
                        004C   208 ;
                        004C   209 ;       NONE
                        004C   210 ;
                        004C   211 ; OUTPUT PARAMETERS:
                        004C   212 ;
                        004C   213 ;       R0      I/O STATUS CODE FROM IOSB
                        004C   214 ;       R1      TST$CHECK_IOSB COMPLETION CODE
                        004C   215 ;       R2      # BYTES TRANSFERRED FROM IOSB
                        004C   216 ;
                        004C   217 ; IMPLICIT OUTPUTS:
                        004C   218 ;
                        004C   219 ;       NONE
                        004C   220 ;
                        004C   221 ; COMPLETION CODES:
                        004C   222 ;
                        004C   223 ;       R1      0 = I/O STATUS CODE IS ABORT (SS$_ABORT) OR
                        004C   224 ;                   STATUS CODE IS CANCEL (SS$_CANCEL) OR
                        004C   225 ;                   STATUS CODE IS REJECT (SS$_REJECT) OR
                        004C   226 ;                   STATUS CODE IS FILE NOT ACCESSED (SS$_FILNOTACC)
                        004C   227 ;               1 = SUCCESS
                        004C   228 ;
                        004C   229 ; SIDE EFFECTS:
                        004C   230 ;
                        004C   231 ;       IF THE I/O STATUS CODE INDICATES FAILURE (EXCEPT AS NOTED ABOVE),
                        004C   232 ;       THE IMAGE IS TERMINATED WITH THE STATUS CODE AS THE EXIT
                        004C   233 ;       COMPLETION CODE.
                        004C   234 ;
                        004C   235 ;--
                        004C   236
                        004C   237 TST$CHECK_IOSB::                              ; CONTROL POINT
52   02 A0   3C         004C   238          MOVZWL  2(R0),R2                    ; EXTRACT BYTE COUNT
     50 60   3C         0050   239          MOVZWL  (R0),R0                     ; EXTRACT I/O STATUS CODE
        AB   11         0053   240          BRB     TST$CHECK_SS                ; CHECK I/O STATUS CODE
```

```
                     0055   242          .SBTTL  TST$QIOW -  NETWORK QIO ROUTINES
                00000055   243          .PSECT  TST$CODE        NOWRT
                     0055   244
                     0055   245 ;++
                     0055   246 ; FUNCTIONAL DESCRIPTION:
                     0055   247 ;
                     0055   248 ;       BOTH TST$QIOW AND TST$QIOAST COMPLETE BUILDING A QIO PARAMETER
                     0055   249 ;       BLOCK AND ISSUE A QIO REQUEST FOR THE ESTABLISHED COMMUNICATIONS
                     0055   250 ;       LINK OR FOR THE ASSOCIATED MAILBOX. THE FUNCTION CODE PARAMETER
                     0055   251 ;       DETERMINES WHICH OF SEVERAL QIO PARAMETER BLOCKS IS USED.
                     0055   252 ;       TST$QIOW ISSUES A $QIOW_G REQUEST AND TST$QIOAST ISSUES A
                     0055   253 ;       $QIO_G WITH AST REQUEST.
                     0055   254 ;
                     0055   255 ; CALLING SEQUENCE:
                     0055   256 ;
                     0055   257 ;       BSB/JSB TST$QIOW
                     0055   258 ;       BSB/JSB TST$QIOAST
                     0055   259 ;
                     0055   260 ; INPUT PARAMETERS:
                     0055   261 ;
                     0055   262 ;       R2      INTERNAL FUNCTION CODE; ALSO SPECIFIES EFN TO USE
                     0055   263 ;       R3      P1 PARAMETER; NOTE: NOT IMPLEMENTED AT PRESENT
                     0055   264 ;       R4      P2 PARAMETER
                     0055   265 ;       R5      ADDRESS OF AST ROUTITNE (FOR TST$QIOAST ONLY)
                     0055   266 ;
                     0055   267 ; IMPLICIT INPUTS:
                     0055   268 ;
                     0055   269 ;       SEVERAL CONTIGUOUS QIO PARAMETER BLOCKS BEGINNING AT TST$PARAMETER.
                     0055   270 ;
                     0055   271 ; OUTPUT PARAMETERS:
                     0055   272 ;
                     0055   273 ;       R0-R1   DESTROYED
                     0055   274 ;
                     0055   275 ; IMPLICIT OUTPUTS:
                     0055   276 ;
                     0055   277 ;       REFERENCED QIO PARAMETER BLOCK (OFFSET FROM TST$PARAMETER) IS
                     0055   278 ;       MODIFIED.
                     0055   279 ;
                     0055   280 ; COMPLETION CODES:
                     0055   281 ;
                     0055   282 ;       NONE
                     0055   283 ;
                     0055   284 ; SIDE EFFECTS:
                     0055   285 ;
                     0055   286 ;       ON COMPLETION OF THE QIO ISSUED BY TST$QIOAST, AN AST ROUTINE
                     0055   287 ;       WILL BE EXECUTED.
                     0055   288 ;
                     0055   289 ;--
                     0055   290
                     0055   291 ;
                     0055   292 ; QIO AND WAIT ROUTINE
                     0055   293 ;
                     0055   294
                     0055   295 TST$QIOW::                                  ; CONTROL POINT
           23   10  0055   296          BSBB    QIO_COMMON                  ; EXECUTE COMMON SET-UP CODE
        14 A0   7C 0057   297          CLRQ    QIO$_ASTADR(R0)             ; ZERO BOTH AST ADDRESS AND
                     005A   298                                              ;  AST PARAMETER LONGWORDS
```

```
                                I 12
                     005A   299          $QIOW_G (R0)                    ; ISSUE THE QIO AND WAIT REQUEST
                     0061   300          CHECK_SS                        ; CHECK STATUS CODE
                  05 0064   301          RSB                             ; EXIT
                     0065   302
                     0065   303 ;
                     0065   304 ; QIO WITH AST ROUTINE
                     0065   305 ;
                     0065   306
                     0065   307
                     0065   308 TST$QIOAST::                             ; CONTROL POINT
                  13 10 0065   309          BSBB    QIO_COMMON              ; EXECUTE COMMON SET-UP CODE
               14 A0 55 D0 0067   310          MOVL    R5,QIO$_ASTADR(R0)      ; UPDATE AST ADDRESS
               18 A0 50 D0 006B   311          MOVL    R0,QIO$_ASTPRM(R0)      ; UPDATE AST PARAMETER WITH
                     006F   312                                          ;  ADDRESS OF THIS PARAMETER BLOCK
                     006F   313          $QIO_G  (R0)                    ; ISSUE QIO WITH AST REQUEST
                     0076   314          CHECK_SS                        ; CHECK STATUS CODE
                  05 0079   315          RSB                             ; EXIT
                     007A   316
                     007A   317 ;
                     007A   318 ; SUBROUTINE THAT PERFORMS COMMON SET-UP FUNCTIONS
                     007A   319 ;
                     007A   320
                     007A   321 QIO_COMMON:                             ; CONTROL POINT
            51 52 0D C5 007A   322          MULL3   #<QIO$_NARGS+1>,R2,R1   ; CALCULATE LONGWORD OFFSET OF
                     007E   323                                          ;  DESIRED QIO PARAMETER BLOCK
                     007E   324                                          ;  FROM THE FIRST PARAMETER BLOCK
         50 0000'CF41 DE 007E   325          MOVAL   W^TST$PARAMETER[R1],R0  ; GET ADDRESS OF PARAMATER BLOCK
      08 A0 0000'CF 3C 0084   326          MOVZWL  W^TST$GW_LINKCHAN,QIO$_CHAN(R0) ; UPDATE CHANNEL #
                  52 D5 008A   327          TSTL    R2                      ; IS DEVICE THE ASSOCIATED MAILBOX?
                  06 12 008C   328          BNEQU   10$                     ; NO
      08 A0 0000'CF 3C 008E   329          MOVZWL  W^TST$GW_MAILCHAN,QIO$_CHAN(R0) ; YES
                     0094   330 ;10$:       MOVL    R3,QIO$_P1(R0)          ; UPDATE BUFFER ADDRESS
            20 A0 54 D0 0094   331 10$:      MOVL    R4,QIO$_P2(R0)          ; UPDATE P2 PARAMETER (EITHER DESC
                     0098   332                                          ;  BLOCK ADDRESS OR BUFFER SIZE)
                  05 0098   333          RSB                             ; EXIT
```

```
                     0099   335          .SBTTL  TST$EXAM_MAIL - EXAMINE MAILBOX MESSAGE
                00000099   336          .PSECT  TST$CODE        NOWRT
                     0099   337
                     0099   338 ;++
                     0099   339 ; FUNCTIONAL DESCRIPTION:
                     0099   340 ;
                     0099   341 ;       TST$EXAM_MAIL DISECTS A MAILBOX MESSAGE INTO ITS VARIOUS
                     0099   342 ;       FIELDS.
                     0099   343 ;
                     0099   344 ; CALLING SEQUENCE:
                     0099   345 ;
                     0099   346 ;       BSB/JSB TST$EXAM_MAIL
                     0099   347 ;
                     0099   348 ; INPUT PARAMETERS:
                     0099   349 ;
                     0099   350 ;       NONE
                     0099   351 ;
                     0099   352 ; IMPLICIT INPUTS:
                     0099   353 ;
                     0099   354 ;       TST$GB_MAILBUF
                     0099   355 ;       TST$GQ_MAILIOSB
                     0099   356 ;
                     0099   357 ; OUTPUT PARAMETERS:
                     0099   358 ;
                     0099   359 ;       R0-R1   DESTROYED
                     0099   360 ;       R6      MAILBOX MESSAGE CODE
                     0099   361 ;       R7      ADDRESS OF RECEIVED MAILBOX DATA LESS HEADER STORED AS A
                     0099   362 ;               COUNTED ASCII STRING
                     0099   363 ;
                     0099   364 ; IMPLICIT OUTPUTS:
                     0099   365 ;
                     0099   366 ;       TST$GW_MAILCODE
                     0099   367 ;       TST$GW_DEV_UNIT
                     0099   368 ;       TST$GT_DEV_NAME
                     0099   369 ;       TST$GT_MAILDATA
                     0099   370 ;
                     0099   371 ; COMPLETION CODES:
                     0099   372 ;
                     0099   373 ;       NONE
                     0099   374 ;
                     0099   375 ; SIDE EFFECTS:
                     0099   376 ;
                     0099   377 ;       NONE
                     0099   378 ;
                     0099   379 ;--
                     0099   380
                     0099   381 TST$EXAM_MAIL::                             ; CONTROL POINT
          3C   BB    0099   382          PUSHR   #^M<R2,R3,R4,R5>           ; SAVE REGISTERS
    51 0000'CF 9E    009B   383          MOVAB   W^TST$GB_MAILBUF,R1        ; GET ADDRESS OF MAILBOX BUFFER
       56   81  3C   00A0   384          MOVZWL  (R1)+,R6                   ; SAVE MAILBOX MESSAGE CODE
0000'CF 56   B0    00A3   385          MOVW    R6,W^TST$GW_MAILCODE       ;
0000'CF 81   B0    00A8   386          MOVW    (R1)+,W^TST$GW_DEV_UNIT    ; STORE DEVICE DEV_UNIT NUMBER
       50   61  9A  00AD   387          MOVZBL  (R1),R0                    ; GET LENGTH OF DEVICE NAME
                   00B0   388                                             ;  COUNTED ASCII STRING
          50   D6  00B0   389          INCL    R0                         ;
0000'CF 61   50  28  00B2   390          MOVC3   R0,(R1),W^TST$GT_DEV_NAME  ; STORE DEVICE NAME STRING
       50   61  9A  00B8   391          MOVZBL  (R1),R0                    ; GET LENGTH OF DATA PORTION OF
```

```
                     00BB                          ; MESSAGE STORED AS A COUNTED STRING
             50  D6  00BB        INCL    R0        ;
  57  0000'CF    9E  00BD        MOVAB   W^TST$GT_MAILDATA,R7  ; GET ADDRESS OF COUNTED STRING
                     00C2                          ;  TO STORE MESSAGE LESS HEADER
  67    61   50  28  00C2        MOVC3   R0,(R1),(R7)          ; STORE MAILBOX MESSAGE LESS HEADER
             3C  BA  00C6        POPR    #^M<R2,R3,R4,R5>      ; RESTORE REGISTERS
                 05  00C8        RSB                           ; EXIT
```

```
                     00C9    400          .SBTTL  TST$FLUSH_MAIL - FLUSH MAILBOX
                000000C9    401          .PSECT  TST$CODE        NOWRT
                     00C9    402
                     00C9    403 ;++
                     00C9    404 ; FUNCTIONAL DESCRIPTION:
                     00C9    405 ;
                     00C9    406 ;       TST$FLUSH_MAIL READS THE MAILBOX UNTIL THERE ARE NO MORE MESSAGES
                     00C9    407 ;       QUEUED FOR IT.
                     00C9    408 ;
                     00C9    409 ; CALLING SEQUENCE:
                     00C9    410 ;
                     00C9    411 ;       BSB/JSB TST$FLUSH_MAIL
                     00C9    412 ;
                     00C9    413 ; INPUT PARAMETERS:
                     00C9    414 ;
                     00C9    415 ;       NONE
                     00C9    416 ;
                     00C9    417 ; IMPLICIT INPUTS:
                     00C9    418 ;
                     00C9    419 ;       TST$GB_MAILBUF
                     00C9    420 ;       TST$GQ_MAILIOSB
                     00C9    421 ;
                     00C9    422 ; OUTPUT PARAMETERS:
                     00C9    423 ;
                     00C9    424 ;       R0-R1   DESTROYED
                     00C9    425 ;
                     00C9    426 ; IMPLICIT OUTPUTS:
                     00C9    427 ;
                     00C9    428 ;       NONE
                     00C9    429 ;
                     00C9    430 ; COMPLETION CODES:
                     00C9    431 ;
                     00C9    432 ;       NONE
                     00C9    433 ;
                     00C9    434 ; SIDE EFFECTS:
                     00C9    435 ;
                     00C9    436 ;       NONE
                     00C9    437 ;
                     00C9    438 ;--
                     00C9    439
                     00C9    440 TST$FLUSH_MAIL::                            ; CONTROL POINT
                     00C9    441         $QIOW_S EFN=#EFN_K_READ_MAIL-       ; ISSUE READ (NOW) TO MAILBOX
                     00C9    442                 CHAN=W^TST$GW_MAILCHAN-     ;
                     00C9    443                 FUNC=#IO$_READVBLK!IO$M_NOW- ;
                     00C9    444                 IOSB=W^TST$GQ_MAILIOSB-     ;
                     00C9    445                 P1=W^TST$GB_MAILBUF-        ;
                     00C9    446                 P2=#TST$K_MAILBUF           ;
0870 8F    50   B1  00F0    447         CMPW    R0,#<SS$_ENDOFFILE&^XFFFF> ; IS IT AN END-OF-FILE?
           09   13  00F5    448         BEQLU   10$                         ; YES
                00F7    449         CHECK_SS                            ; CHECK STATUS CODE
     0002'CF    B5  00FA    450         TSTW    W^TST$GQ_MAILIOSB+2         ; DID WE RECEIVE ANYTHING?
           C9   12  00FE    451         BNEQU   TST$FLUSH_MAIL              ; YES, READ AGAIN
                05  0100    452 10$:    RSB                                 ; EXIT
```

```
                     0101    454          .SBTTL  TST$PPRINT_FAO - PRINT OUTPUT FROM FAO
                00000101    455          .PSECT  TST$CODE        NOWRT
                     0101    456
                     0101    457 ;++
                     0101    458 ; FUNCTIONAL DESCRIPTION:
                     0101    459 ;
                     0101    460 ;       TST$PRINT_FAO OUTPUTS THE BUFFER FORMATTED BY FAO TO THE PRINT
                     0101    461 ;       DEVICE.
                     0101    462 ;
                     0101    463 ; CALLING SEQUENCE:
                     0101    464 ;
                     0101    465 ;       BSB/JSB TST$PRINT_FAO
                     0101    466 ;
                     0101    467 ; INPUT PARAMETERS:
                     0101    468 ;
                     0101    469 ;       NONE
                     0101    470 ;
                     0101    471 ; IMPLICIT INPUTS:
                     0101    472 ;
                     0101    473 ;       TST$GB_PRTBUF
                     0101    474 ;       TST$GW_PRTLEN
                     0101    475 ;
                     0101    476 ; OUTPUT PARAMETERS:
                     0101    477 ;
                     0101    478 ;       R0-R1   DESTROYED
                     0101    479 ;
                     0101    480 ; IMPLICIT OUTPUTS:
                     0101    481 ;
                     0101    482 ;       PRTRAB IS UPDATED
                     0101    483 ;
                     0101    484 ; COMPLETION CODES:
                     0101    485 ;
                     0101    486 ;       NONE
                     0101    487 ;
                     0101    488 ; SIDE EFFECTS:
                     0101    489 ;
                     0101    490 ;       NONE
                     0101    491 ;
                     0101    492 ;--
                     0101    493
                     0101    494 TST$PRINT_FAO::                             ; CONTROL POINT
          0000'CF    B0  0101    495          MOVW    W^TST$GW_PRTLEN,-       ; UPDATE BUFFER SIZE IN PRINT RAB
          0022'CF        0105    496                  W^TST$PRTRAB+RAB$W_RSZ  ;
                         0108    497          $PUT    RAB=W^TST$PRTRAB        ; OUTPUT THE RECORD
                         0113    498          CHECK_RMS                       ; CHECK COMPLETION CODE
                     05  0116    499          RSB                             ; EXIT
                         0117    500 TST$FAOOUT::
                   0000  0117    501          .WORD   0                       ;FORMAT COUNTED FAO STRING
          5E    F8 AE    DE  0119    502          MOVAL   -8(SP),SP               ;ALLOCATE SPACE FOR DESCRIPTOR
          6E    04 BC    9A  011D    503          MOVZBL  @4(AP),(SP)             ;CONTROL STRING LENGTH
04 AE     04 AC    01    C1  0121    504          ADDL3   #1,4(AP),4(SP)          ;ADDRESS CONTROL STRING PORTION
                         0127    505          $FAOL_S CTRSTR=(SP)-
                         0127    506                  OUTLEN=W^TST$GW_PRTLEN-
                         0127    507                  OUTBUF=W^TST$GQ_PRTBUF-
                         0127    508                  PRMLST=8(AP)
                FFC3     30  013B    509          BSBW    W^TST$PRINT_FAO         ;PRINT FAO STRING
                         04  013E    510          RET
```

```
                     013F   512          .SBTTL  TST$DISPLAY_MSG - DISPLAY MESSAGE
                0000013F    513          .PSECT  TST$CODE        NOWRT
                     013F   514
                     013F   515 ;++
                     013F   516 ; FUNCTIONAL DESCRIPTION:
                     013F   517 ;
                     013F   518 ;       TST$DISPLAY_MSG DISPLAYS THE MESSAGE LENGTH (IN BYTES) AND UP TO
                     013F   519 ;       THE SPECIFIED NUMBER OF BYTES OF THE MESSAGE BUFFER IN HEXADECIMAL.
                     013F   520 ;
                     013F   521 ; CALLING SEQUENCE:
                     013F   522 ;
                     013F   523 ;       CALL    #4,TST$DISPLAY_MSG
                     013F   524 ;
                     013F   525 ; INPUT PARAMETERS:
                     013F   526 ;
                     013F   527 ;       4(AP)   MAXIMUM NUMBER OF BYTES TO DISPLAY
                     013F   528 ;       8(AP)   TRANSMIT/RECEIVE INDICATOR (0/1)
                     013F   529 ;       12(AP)  ADDRESS OF THE MESSAGE
                     013F   530 ;       16(AP)  SIZE OF THE MESSAGE IN BYTES
                     013F   531 ;
                     013F   532 ; IMPLICIT INPUTS:
                     013F   533 ;
                     013F   534 ;       NONE
                     013F   535 ;
                     013F   536 ; OUTPUT PARAMETERS:
                     013F   537 ;
                     013F   538 ;       R0-R1   DESTROYED
                     013F   539 ;
                     013F   540 ; IMPLICIT OUTPUTS:
                     013F   541 ;
                     013F   542 ;       NONE
                     013F   543 ;
                     013F   544 ; COMPLETION CODES:
                     013F   545 ;
                     013F   546 ;       NONE
                     013F   547 ;
                     013F   548 ; SIDE EFFECTS:
                     013F   549 ;
                     013F   550 ;       NONE
                     013F   551 ;
                     013F   552 ;--
                     013F   553
                0004 013F   554          .ENTRY  TST$DISPLAY_MSG,^M<R2>  ; ENTRY POINT
                     0141   555
                     0141   556 ;
                     0141   557 ; DETERMINE NUMBER OF BYTES TO DISPLAY
                     0141   558 ;
                     0141   559
 50   04 AC   D0     0141   560          MOVL    4(AP),R0                ; GET MAX #BYTES TO DISPLAY
        48   13     0145   561          BEQL    50$                     ; IF NONE, WE'RE FINISHED
 50   10 AC   D1     0147   562          CMPL    16(AP),R0               ; IS MESSAGE SIZE GEQ MAX COUNT?
        04   18     014B   563          BGEQ    10$                     ; YES
 50   10 AC   D0     014D   564          MOVL    16(AP),R0               ; NO, USE ACTUAL MESSAGE SIZE
    51   50   D0     0151   565 10$:     MOVL    R0,R1                   ; SAVE COUNT
                     0154   566
                     0154   567 ;
                     0154   568 ; CONSTRUCT PARAMETER LIST FOR FAO ON THE STACK
```

```
                     0154   569 ;
                     0154   570
52   0C AC    D0     0154   571          MOVL    12(AP),R2                       ; GET MESSAGE ADDRESS
  7E    82    9A     0158   572 20$:     MOVZBL  (R2)+,-(SP)                     ; PUT EACH CHARACTER IN LIST
    FA 50    F5     015B   573          SOBGTR  R0,20$                          ; CONTINUE UNTIL DONE
       51    DD     015E   574          PUSHL   R1                              ; PUT #BYTES TO CONVERT IN LIST
    10 AC    DD     0160   575          PUSHL   16(AP)                          ; PUT MESSAGE SIZE IN LIST
  06 08 AC    E8     0163   576          BLBS    8(AP),30$                       ; IS THIS A XMIT OR RECV?
  0000'CF    9F     0167   577          PUSHAB  W^TST$GT_XMIT                   ; PUT ADDRESS OF TEXT IN LIST
       04    11     016B   578          BRB     40$                             ;
  0000'CF    9F     016D   579 30$:     PUSHAB  W^TST$GT_RECV                   ; PUT ADDRESS OF TEXT IN LIST
  51   5E    D0     0171   580 40$:     MOVL    SP,R1                           ; GET ADDRESS OF FAO PARAMETER LIST
                     0174   581
                     0174   582 ;
                     0174   583 ; FORMAT AND PRINT THE MESSAGE
                     0174   584 ;
                     0174   585
                     0174   586          $FAOL_S CTRSTR=W^TST$GQ_DISPLAY- ; FORMAT MESSAGE
                     0174   587                  OUTLEN=W^TST$GW_PRTLEN-  ;
                     0174   588                  OUTBUF=W^TST$GQ_PRTBUF-  ;
                     0174   589                  PRMLST=(R1)              ;
                     0189   590          CHECK_SS                                ; CHECK STATUS CODE
     FF72    30     018C   591          BSBW    TST$PRINT_FAO                   ; PRINT MESSAGE
                     018F   592
                     018F   593 ;
                     018F   594 ; "RET" INSTRUCTION WILL ADJUST SP TO THAT FAO PARAMETER LIST
                     018F   595 ; THAT WAS CONSTRUCTED ON THE STACK IS ELIMINATED.
                     018F   596 ;
                     018F   597
              04     018F   598 50$:     RET                                     ; EXIT
```

```
                     0190   600          .SBTTL  TST$STANDARD - MOVE STANDARD DATA PATTERN
                00000190   601          .PSECT  TST$CODE         NOWRT
                     0190   602
                     0190   603 ;++
                     0190   604 ; FUNCTIONAL DESCRIPTION:
                     0190   605 ;
                     0190   606 ;       TST$STANDARD FILLS THE DESIGNATED BUFFER WITH REPETITIONS OF
                     0190   607 ;       THE "STANDARD" DATA PATTERN.
                     0190   608 ;
                     0190   609 ; CALLING SEQUENCE:
                     0190   610 ;
                     0190   611 ;       BSB/JSB TST$STANDARD
                     0190   612 ;
                     0190   613 ; INPUT PARAMETERS:
                     0190   614 ;
                     0190   615 ;       R3      ADDRESS OF THE BUFFER
                     0190   616 ;       R4      SIZE OF THE BUFFER IN BYTES
                     0190   617 ;
                     0190   618 ; IMPLICIT INPUTS:
                     0190   619 ;
                     0190   620 ;       TST$GT_STANDARD = COUNTED ASCII STRING OF STANDARD DATA PATTERN
                     0190   621 ;
                     0190   622 ; OUTPUT PARAMETERS:
                     0190   623 ;
                     0190   624 ;       R0-R1   DESTROYED
                     0190   625 ;
                     0190   626 ; IMPLICIT OUTPUTS:
                     0190   627 ;
                     0190   628 ;       NONE
                     0190   629 ;
                     0190   630 ; COMPLETION CODES:
                     0190   631 ;
                     0190   632 ;       NONE
                     0190   633 ;
                     0190   634 ; SIDE EFFECTS:
                     0190   635 ;
                     0190   636 ;       NONE
                     0190   637 ;
                     0190   638 ;--
                     0190   639
                     0190   640 TST$STANDARD::                                  ; CONTROL POINT
          03FC 8F    BB  0190   641          PUSHR   #^M<R2,R3,R4,R5,R6,R7,R8,R9> ; SAVE REGISTERS
     56   0000'CF    DE  0194   642          MOVAL   W^TST$GT_STANDARD,R6        ; GET ADDRESS OF COUNTED
                     0199   643                                                  ;  STANDARD DATA STRING
          57   86    9A  0199   644          MOVZBL  (R6)+,R7                    ; GET SIZE OF STANDARD DATA STRING
               55    D4  019C   645          CLRL    R5                          ; DOUBLE PRECISION DIVISION FOLLOWS
                     019E   646                                                  ;  I.E., {R4,R5} / R7 = R8 R R9
59   58   54   57    7B  019E   647          EDIV    R7,R4,R8,R9                 ; PUT LOOP COUNT IN R8
               07    13  01A3   648          BEQLU   20$                         ; IS BUFFER SMALLER THAN STD PATTERN?
     63   66   57    28  01A5   649 10$:     MOVC3   R7,(R6),(R3)                ; NO, COPY STANDARD DATA PATTERN
            F9 58    F5  01A9   650          SOBGTR  R8,10$                      ; WILL PATTERN FIT?
     63   66   59    28  01AC   651 20$:     MOVC3   R9,(R6),(R3)                ; NO, FILL REMAINDER OF BUFFER
          03FC 8F    BA  01B0   652          POPR    #^M<R2,R3,R4,R5,R6,R7,R8,R9> ; RESTORE REGISTERS
                     05  01B4   653          RSB                                 ; EXIT
                         01B5   654          .END
```

```
$$.TMP1                         = 00000001
$$.TMP2                         = 000000CF
$$ARGS                          = 0000000C
$$T1                            = 00000001
C                                 00000000 RG    02
EFN_K_READ_MAIL                 = 00000000
IO$M_NOW                          ******** X   02
IO$_READVBLK                      ******** X   02
K_LIST_MEB                      = 00000000
QIO$_ASTADR                     = 00000014
QIO$_ASTPRM                     = 00000018
QIO$_CHAN                       = 00000008
QIO$_EFN                        = 00000004
QIO$_FUNC                       = 0000000C
QIO$_IOSB                       = 00000010
QIO$_NARGS                      = 0000000C
QIO$_P1                         = 0000001C
QIO$_P2                         = 00000020
QIO$_P3                         = 00000024
QIO$_P4                         = 00000028
QIO$_P5                         = 0000002C
QIO$_P6                         = 00000030
QIO_COMMON                        0000007A R     02
RAB$W_RSZ                       = 00000022
RMS$_EOF                        = 0001827A
RMS$_TMO                        = 000181B0
SS$_ABORT                       = 0000002C
SS$_CANCEL                      = 00000830
SS$_ENDOFFILE                   = 00000870
SS$_FILNOTACC                   = 000000AC
SS$_LINKABORT                   = 000020E4
SS$_NORMAL                      = 00000001
SS$_REJECT                      = 00000294
SYS$EXIT                          ******** GX    02
SYS$FAOL                          ******** GX    02
SYS$PUT                           ******** GX    02
SYS$QIO                           ******** GX    02
SYS$QIOW                          ******** GX    02
TST$CHECK_IOSB                    0000004C RG    02
TST$CHECK_RMS                     0000002C RG    02
TST$CHECK_SS                      00000000 RG    02
TST$DISPLAY_MSG                   0000013F RG    02
TST$EXAM_MAIL                     00000099 RG    02
TST$FAOOUT                        00000117 RG    02
TST$FLUSH_MAIL                    000000C9 RG    02
TST$GB_MAILBUF                    ********   X   02
TST$GQ_DISPLAY                    ********   X   02
TST$GQ_MAILIOSB                   ********   X   02
TST$GQ_PRTBUF                     ********   X   02
TST$GT_DEV_NAME                   ********   X   02
TST$GT_MAILDATA                   ********   X   02
TST$GT_RECV                       ********   X   02
TST$GT_STANDARD                   ********   X   02
TST$GT_XMIT                       ********   X   02
TST$GW_DEV_UNIT                   ********   X   02
TST$GW_LINKCHAN                   ********   X   02
TST$GW_MAILCHAN                   ********   X   02
TST$GW_MAILCODE                   ********   X   02
TST$GW_PRTLEN                     ********   X   02
TST$K_MAILBUF                     ********   X   02
TST$PARAMETER                     ********   X   02
TST$PRINT_FAO                     00000101 RG    02
TST$PRTRAB                        ********    X  02
TST$QIOAST                        00000065 RG    02
TST$QIOW                          00000055 RG    02
TST$STANDARD                      00000190 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	000001B5 (437.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.12	00:00:00.60
Command processing	142	00:00:00.79	00:00:04.50
Pass 1	299	00:00:08.94	00:00:23.30
Symbol table sort	0	00:00:01.08	00:00:01.23
Pass 2	115	00:00:02.33	00:00:04.64
Symbol table output	9	00:00:00.11	00:00:00.09
Psect synopsis output	2	00:00:00.01	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	602	00:00:13.40	00:00:34.42

The working set limit was 1350 pages.
47661 bytes (94 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 814 non-local and 13 local symbols.
716 source lines were read in Pass 1, producing 18 object records in Pass 2.
29 pages of virtual memory were used to define 27 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[DTSDTR.OBJ]DTSDTR.MLB;1	3
_255DUA28:[SYSLIB]STARLET.MLB;2	19
TOTALS (all libraries)	22

989 GETS were required to define 22 macros.

There were no errors, warnings or information messages.

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

XWDRIVER
LIS

DTGLOBAL
LIS

DTDEFINE
LIS

DTRMAIN
LIS

DTRAST
LIS

DTPREFIX
MAR

DTSDTR

DTCOMMON
LIS

DTRECV
MAP

DTSEND
MAP

DTMACROS
MAR