RRRRRRRRRRR TTTTTTTTTTTTTTT PPPPPPPPPPP AAAAAAAAA DDDDDDDDDD
RRRRRRRRRRR TTTTTTTTTTTTTTT PPPPPPPPPPP AAAAAAAAA DDDDDDDDDD
RRRRRRRRRRR TTTTTTTTTTTTTTT PPPPPPPPPPP AAAAAAAAA DDDDDDDDDD
RRR RRR TTT PPP PPP AAA AAA DDD DDD
RRR RRR TTT PPP PPP AAA AAA DDD DDD
RRR RRR TTT PPP PPP AAA AAA DDD DDD
RRR RRR TTT PPP PPP AAA AAA DDD DDD
RRR RRR TTT PPP PPP AAA AAA DDD DDD
RRRRRRRRRRR TTT PPPPPPPPPPP AAA AAA DDD DDD
RRRRRRRRRRR TTT PPPPPPPPPPP AAA AAA DDD DDD
RRRRRRRRRRR TTT PPPPPPPPPPP AAA AAA DDD DDD
RRR RRR TTT PPP AAAAAAAAAAAAAAA DDD DDD
RRR RRR TTT PPP AAAAAAAAAAAAAAA DDD DDD
RRR RRR TTT PPP AAAAAAAAAAAAAAA DDD DDD
RRR RRR TTT PPP AAA AAA DDD DDD
RRR RRR TTT PPP AAA AAA DDD DDD
RRR RRR TTT PPP AAA AAA DDD DDD
RRR RRR TTT PPP AAA AAA DDDDDDDDDDD
RRR RRR TTT PPP AAA AAA DDDDDDDDDDD
RRR RRR TTT PPP AAA AAA DDDDDDDDDDD

FILEID**RTDTE

```
RRRRRRRR    TTTTTTTTTT  DDDDDDDD    TTTTTTTTTT  EEEEEEEEEE
RRRRRRRR    TTTTTTTTTT  DDDDDDDD    TTTTTTTTTT  EEEEEEEEEE
RR      RR      TT      DD      DD      TT      EE
RR      RR      TT      DD      DD      TT      EE
RR      RR      TT      DD      DD      TT      EE
RR      RR      TT      DD      DD      TT      EE
RRRRRRRR        TT      DD      DD      TT      EEEEEEEE
RRRRRRRR        TT      DD      DD      TT      EEEEEEEE
RR  RR          TT      DD      DD      TT      EE
RR  RR          TT      DD      DD      TT      EE
RR    RR        TT      DD      DD      TT      EE              ....
RR    RR        TT      DD      DD      TT      EE              ....
RR      RR      TT      DDDDDDDD        TT      EEEEEEEEEE      ....
RR      RR      TT      DDDDDDDD        TT      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
```

```
;     1          0001  0 MODULE RTDTE ( IDENT = 'V04-000',
;     2          0002  0                 ADDRESSING_MODE(EXTERNAL=GENERAL)
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !**************************************************************************
;     8          0008  1 !*                                                                        *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                   *
;    12          0012  1 !*                                                                        *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18          0018  1 !*  TRANSFERRED.                                                           *
;    19          0019  1 !*                                                                        *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22          0022  1 !*  CORPORATION.                                                           *
;    23          0023  1 !*                                                                        *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !**************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:
;    33          0033  1 !
;    34          0034  1 !       SET HOST/LINE=ttcn:
;    35          0035  1 !
;    36          0036  1 ! ABSTRACT:
;    37          0037  1 !
;    38          0038  1 !       Provides data terminal equipment (DTE) or terminal emulation
;    39          0039  1 !       routines in SET HOST.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:
;    42          0042  1 !
;    43          0043  1 !       VAX/VMS user mode.
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 ! AUTHOR: Jake VanNoy, CREATION DATE: 9-Jan-1984
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V03-001 JLV0365         Jake VanNoy             11-JUL-1984
;    52          0052  1 !               Bug fixes and better messages.
;    53          0053  1 !
;    54          0054  1 !**
```

```
;    56          0055  1
;    57          0056  1 !
;    58          0057  1 ! Include files
;    59          0058  1 !
;    60          0059  1 LIBRARY 'SYS$LIBRARY:STARLET';                  ! VAX/VMS system definitions
;    61          0060  1
;    62          0061  1 REQUIRE 'OBJ$:RTDEF';                           ! rtpad stuff
;    63          0612  1
;    64          0613  1 REQUIRE 'SYS$LIBRARY:UTILDEF';                  ! Private VAX/VMS macros
;    65          0789  1 !
;    66          0790  1 ! Table of contents
;    67          0791  1 !
;    68          0792  1 FORWARD ROUTINE
;    69          0793  1     TERM$EMULATE,                               ! Main routine
;    70          0794  1     INITIALIZE,                                 ! initialize
;    71          0795  1     DIAL_NUMBER,                                ! autodial number
;    72          0796  1     TRANSMIT,                                   ! do transfers
;    73          0797  1     READ_COMMAND_KB:        NOVALUE,            ! read command keyboard
;    74          0798  1     READ_KB_AST:            NOVALUE,            ! read command keyboard AST
;    75          0799  1     READ_PORT_MBX:          NOVALUE,            ! read port mailbox
;    76          0800  1     READ_MBX_AST:           NOVALUE,            ! read port mailbox AST
;    77          0801  1     READ_SYSINPUT,                              ! read from SYS$INPUT
;    78          0802  1     READ_PORT_TIMED:        NOVALUE,            ! read from port with timed read
;    79          0803  1     READ_PORT_AST:          NOVALUE,            ! ast from port with timed read
;    80          0804  1     READ_DELAY_AST:         NOVALUE,            ! set timer ast routine
;    81          0805  1     SET_PASSTHRU_MODE,                          ! set terminal mode
;    82          0806  1     WRITE_COMMAND:          NOVALUE,            ! start write to command channel
;    83          0807  1     WRITE_COMMAND_AST:      NOVALUE,            ! ast for write to Command channel
;    84          0808  1     ALLOCATE_BUF,                               ! allocate buffer
;    85          0809  1     DEALLOCATE_BUF:         NOVALUE,            ! deallocate buffer
;    86          0810  1     SENSE_MODE,                                 ! sense characteristics
;    87          0811  1     dump_stats:             NOVALUE,            ! %%
;    88          0812  1     EXIT_ROUTINE;                               ! exit handler
;    89          0813  1
;    90          0814  1 !
;    91          0815  1 ! Literals
;    92          0816  1 !
;    93          0817  1 LITERAL
;    94          0818  1     w_riosb_status      = 0,
;    95          0819  1     w_riosb_datasize    = 1,
;    96          0820  1     b_riosb_termchar    = 4,
;    97          0821  1     b_riosb_termsize    = 6,
;    98          0822  1     char$c_cr           = 13,
;    99          0823  1     char$c_lf           = 10,
;   100          0824  1     log_bufsiz = 512,
;   101          0825  1     rte$c_portread_max  = 2;
;   102          0826  1 !
;   103          0827  1 ! Macros
;   104          0828  1 !
;   105          0829  1
;   106          0830  1 MACRO
;   107        M 0831  1     quit_if_error(command) =
;   108        M 0832  1         BEGIN
;   109        M 0833  1         LOCAL
;   110        M 0834  1             status;
;   111        M 0835  1
;   112        M 0836  1         status = command;
```

```
;  113          M 0837  1          IF NOT .status
;  114          M 0838  1          THEN
;  115          M 0839  1              BEGIN
;  116          M 0840  1              WAKEFLAG = 1;
;  117          M 0841  1              $WAKE ();
;  118          M 0842  1              RETSTATUS = .status;
;  119          M 0843  1              RETURN .status;
;  120          M 0844  1              END;
;  121            0845  1          END%;
;  122            0846  1
;  123            0847  1  MACRO
;  124          M 0848  1      quit (status) =
;  125          M 0849  1          BEGIN
;  126          M 0850  1          WAKEFLAG = 1;
;  127          M 0851  1          $WAKE ();
;  128          M 0852  1          RETSTATUS = status;
;  129          M 0853  1          RETURN status;
;  130            0854  1          END%;
;  131            0855  1
;  132            0856  1  MACRO
;  133          M 0857  1          PRINT (string) =        ! print debug data
;  134          M 0858  1          BEGIN
;  135          M 0859  1          LOCAL status;
;  136          M 0860  1
;  137          M 0861  1          outdesc [0] = 132;
;  138          M 0862  1          status = SYS$FAO (string, outdesc, outdesc, %remaining);
;  139          M 0863  1
;  140          M 0864  1          IF NOT .status THEN signal (.status);
;  141          M 0865  1          LIB$PUT_OUTPUT (outdesc);
;  142            0866  1          END%;
;  143            0867  1
;  144            0868  1  !
;  145            0869  1  ! External Routines
;  146            0870  1  !
;  147            0871  1  EXTERNAL ROUTINE
;  148            0872  1      LIB$FIND_IMAGE_SYMBOL,
;  149            0873  1      STR$CONCAT,
;  150            0874  1      LIB$PUT_OUTPUT,      ! %% for stats
;  151            0875  1      SYS$FAO,             ! %% for stats
;  152            0876  1      debug_ast,           ! ***
;  153            0877  1      CLI$PRESENT,         ! cli routine
;  154            0878  1      CLI$GET_VALUE,       ! cli routine
;  155            0879  1      SYS$SETIMR,
;  156            0880  1      RTLOG$WRITE_STRING,
;  157            0881  1      LIB$GET_VM,
;  158            0882  1      LIB$FREE_VM,
;  159            0883  1      LIB$SIGNAL,
;  160            0884  1      LIB$ASN_WTH_MBX;                    ! assign channel with assoc. mailbox
;  161            0885  1  !
;  162            0886  1  ! External Storage
;  163            0887  1  !
;  164            0888  1  EXTERNAL
;  165            0889  1      RTLOG_FLAGS,
;  166            0890  1      CTERM_FLAG,
;  167            0891  1      INDFLAG,
;  168            0892  1      WAKEFLAG,
;  169            0893  1      RETSTATUS;
```

```
;  170     0894  1
;  171     0895  1 !
;  172     0896  1 ! External Literals
;  173     0897  1 !
;  174     0898  1 EXTERNAL LITERAL
;  175     0899  1     REM$_TOEXIT,
;  176     0900  1     CLI$_ABSENT;
;  177     0901  1 !
;  178     0902  1 ! Builtin Functions
;  179     0903  1 !
;  180     0904  1 BUILTIN
;  181     0905  1     REMQUE,
;  182     0906  1     INSQUE;
;  183     0907  1 !
;  184     0908  1 ! Own Storage
;  185     0909  1 !
;  186     0910  1 OWN
;  187     0911  1
;  188     0912  1     !
;  189     0913  1     ! Counters
;  190     0914  1     !
;  191     0915  1     max_port_count:     INITIAL(0),     ! max port cnt
;  192     0916  1     cnt_rp_timed:       INITIAL(0),     ! calls to READ_PORT_TIMED
;  193     0917  1     cnt_rp_char:        INITIAL(0),     ! total number of chars read
;  194     0918  1     cnt_rp_zero:        INITIAL(0),     ! total of read_port_timed with no char
;  195     0919  1     cnt_rp_timer:       INITIAL (0),    !
;  196     0920  1     cnt_rp_uns:         INITIAL (0),    !
;  197     0921  1     cnt_rp_ast:         INITIAL (0),    !
;  198     0922  1     cnt_wr_ast:         INITIAL (0),    !
;  199     0923  1     !
;  200     0924  1     ! for real work
;  201     0925  1     !
;  202     0926  1
;  203     0927  1     log_buffer:         VECTOR [log_bufsiz,BYTE],
;  204     0928  1     log_count:          INITIAL (0),
;  205     0929  1     command_chan:       WORD,
;  206     0930  1     port_chan:          WORD,
;  207     0931  1     port_mbx_chan:      WORD,
;  208     0932  1     port_count:         INITIAL (0),
;  209     0933  1     write_active_count: INITIAL (0),                    ! counter
;  210     0934  1     timer_pending:      INITIAL (false),                ! boolean
;  211     0935  1     read_pending:       INITIAL (false),                ! boolean
;  212     0936  1     msgsiz:             INITIAL (RTE$C_BUFLEN),
;  213     0937  1     delay_time:         VECTOR [2],                     ! quadword delta time
;  214     0938  1     write_command_q:    VECTOR [2],
;  215     0939  1     command_char:       VECTOR [3],
;  216     0940  1     port_char:          VECTOR [3],
;  217     0941  1     exit_status,
;  218     0942  1     exit_block:         VECTOR [4]
;  219     0943  1                         INITIAL(0,exit_routine,
;  220     0944  1                                 1,exit_status);
;  221     0945  1
```

```
;  223     0946  1 GLOBAL ROUTINE TERM$EMULATE (port_name: REF VECTOR, cmd_chan: WORD) =
;  224     0947  1
;  225     0948  1 !++
;  226     0949  1 !
;  227     0950  1 ! FUNCTIONAL DESCRIPTION:
;  228     0951  1 !
;  229     0952  1 !       main routine
;  230     0953  1 !
;  231     0954  1 !
;  232     0955  1 ! CALLING SEQUENCE:
;  233     0956  1 !
;  234     0957  1 ! INPUT PARAMETERS:
;  235     0958  1 !       port_name:      descriptor of port name (from /LINE= qualfier)
;  236     0959  1 !       command_chan:   channel to SYS$COMMAND terminal
;  237     0960  1 !
;  238     0961  1 ! IMPLICIT INPUTS:
;  239     0962  1 !
;  240     0963  1 !       INDFLAG         - set to 1 if SYS$INPUT is a file
;  241     0964  1 !
;  242     0965  1 ! OUTPUT PARAMETERS:
;  243     0966  1 !       NONE
;  244     0967  1 !
;  245     0968  1 ! IMPLICIT OUTPUTS:
;  246     0969  1 !       NONE
;  247     0970  1 !
;  248     0971  1 ! ROUTINE VALUE:
;  249     0972  1 !       NONE
;  250     0973  1 !
;  251     0974  1 ! SIDE EFFECTS:
;  252     0975  1 !       NONE
;  253     0976  1 !
;  254     0977  1 !--
;  255     0978  1
;  256     0979  2 BEGIN
;  257     0980  2
;  258     0981  2 command_chan = .cmd_chan;                       ! save command channel
;  259     0982  2
;  260     0983  2 Return_if_error (initialize(.port_name));       ! assign channels, etc.
;  261     0984  2
;  262     0985  2 Return_if_error (dial_number());                ! dial phone number
;  263     0986  2
;  264     0987  2 !
;  265     0988  2 ! tell user how to exit
;  266     0989  2 !
;  267     0990  2 Lib$signal (REM$_TOEXIT, 1, %ASCID '^\' );
;  268     0991  2
;  269     0992  2 Return_if_error (transmit ());                  ! do real work
;  270     0993  2
;  271     0994  2 RETURN (1);
;  272     0995  2
;  273     0996  1 END;                                    ! End of routine

                                                        .TITLE  RTDTE
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2
```

```
00  00  5C  5E  0000C P.AAB:  .ASCII  \^\<92><0><0>                          ;
      010E0002  00004 P.AAA:  .LONG   17694722                               ;
      00000000' 00008         .ADDRESS P.AAB                                 ;

                              .PSECT  $OWN$,NOEXE,2

      00000000  00000 MAX_PORT_COUNT:
                              .LONG   0                                      ;
      00000000  00004 CNT_RP_TIMED:
                              .LONG   0                                      ;
      00000000  00008 CNT_RP_CHAR:
                              .LONG   0                                      ;
      00000000  0000C CNT_RP_ZERO:
                              .LONG   0                                      ;
      00000000  00010 CNT_RP_TIMER:
                              .LONG   0                                      ;
      00000000  00014 CNT_RP_UNS:
                              .LONG   0                                      ;
      00000000  00018 CNT_RP_AST:
                              .LONG   0                                      ;
      00000000  0001C CNT_WR_AST:
                              .LONG   0                                      ;
                00020 LOG_BUFFER:
                              .BLKB   512
      00000000  00220 LOG_COUNT:
                              .LONG   0                                      ;
                00224 COMMAND_CHAN:
                              .BLKB   2
                00226 PORT_CHAN:
                              .BLKB   2
                00228 PORT_MBX_CHAN:
                              .BLKB   2
                0022A         .BLKB   2
00000000  0022C PORT_COUNT:
                              .LONG   0                                      ;
00000000  00230 WRITE_ACTIVE_COUNT:
                              .LONG   0                                      ;
00000000  00234 TIMER_PENDING:
                              .LONG   0                                      ;
00000000  00238 READ_PENDING:
                              .LONG   0                                      ;
00000050  0023C MSGSIZ: .LONG   80                                           ;
                00240 DELAY_TIME:
                              .BLKB   8
                00248 WRITE_COMMAND_Q:
                              .BLKB   8
                00250 COMMAND_CHAR:
                              .BLKB   12
                0025C PORT_CHAR:
                              .BLKB   12
                00268 EXIT_STATUS:
                              .BLKB   4
00000000  0026C EXIT_BLOCK:
                              .LONG   0                                      ;
00000000V 00270         .ADDRESS EXIT_ROUTINE                                ;
00000001  00274         .LONG   1                                            ;
```

```
                                00000000' 00278         .ADDRESS EXIT_STATUS                                    ;

                                                        .EXTRN  LIB$FIND_IMAGE_SYMBOL
                                                        .EXTRN  STR$CONCAT, LIB$PUT_OUTPUT
                                                        .EXTRN  SYS$FAO, DEBUG_AST
                                                        .EXTRN  CLI$PRESENT, CLI$GET_VALUE
                                                        .EXTRN  SYS$SETIMR, RTLOG$WRITE_STRING
                                                        .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                        .EXTRN  LIB$SIGNAL, LIB$ASN_WTH_MBX
                                                        .EXTRN  RTLOG_FLAGS, CTERM_FLAG
                                                        .EXTRN  INDFLAG, WAKEFLAG
                                                        .EXTRN  RETSTATUS, REM$_TOEXIT
                                                        .EXTRN  CLI$_ABSENT

                                                        .PSECT  $CODE$,NOWRT,2

                                    0000 00000          .ENTRY  TERM$EMULATE, Save nothing                      ; 0946
                        0000'  CF       08  AC  B0 00002        MOVW    CMD_CHAN, COMMAND_CHAN                  ; 0981
                                        04  AC  DD 00008        PUSHL   PORT_NAME                               ; 0983
                        0000V  CF           01  FB 0000B        CALLS   #1, INITIALIZE                          ;
                               26           50  E9 00010        BLBC    STATUS, 1$                              ;
                        0000V  CF           00  FB 00013        CALLS   #0, DIAL_NUMBER                         ; 0985
                               1E           50  E9 00018        BLBC    STATUS, T$                              ;
                                    0000'  CF  9F 0001B        PUSHAB  P.AAA                                   ; 0990
                                           01  DD 0001F        PUSHL   #1                                      ;
                                00000000G  8F  DD 00021        PUSHL   #REM$_TOEXIT                            ;
                    00000000G  00          03  FB 00027        CALLS   #3, LIB$SIGNAL                          ;
                        0000V  CF          00  FB 0002E        CALLS   #0, TRANSMIT                            ; 0992
                               03          50  E9 00033        BLBC    STATUS, 1$                              ;
                               50          01  D0 00036        MOVL    #1, R0                                  ; 0994
                                           04  00039 1$:       RET                                             ; 0996

; Routine Size:  58 bytes,    Routine Base:  $CODE$ + 0000

;  274          0997  1
```

```
;  276     0998  1 ROUTINE INITIALIZE (port_name: REF VECTOR) =
;  277     0999  1
;  278     1000  1 !++
;  279     1001  1 !
;  280     1002  1 ! FUNCTIONAL DESCRIPTION:
;  281     1003  1 !
;  282     1004  1 !       Assign a channel to the port terminal.
;  283     1005  1 !
;  284     1006  1 ! CALLING SEQUENCE:
;  285     1007  1 !
;  286     1008  1 !       INITIALIZE();
;  287     1009  1 !
;  288     1010  1 ! INPUT PARAMETERS:
;  289     1011  1 !       NONE
;  290     1012  1 !
;  291     1013  1 ! IMPLICIT INPUTS:
;  292     1014  1 !       NONE
;  293     1015  1 !
;  294     1016  1 ! OUTPUT PARAMETERS:
;  295     1017  1 !       NONE
;  296     1018  1 !
;  297     1019  1 ! IMPLICIT OUTPUTS:
;  298     1020  1 !
;  299     1021  1 !       port_chan       - port channel
;  300     1022  1 !
;  301     1023  1 ! ROUTINE VALUE:
;  302     1024  1 !       NONE
;  303     1025  1 !
;  304     1026  1 ! SIDE EFFECTS:
;  305     1027  1 !       NONE
;  306     1028  1 !
;  307     1029  1 !--
;  308     1030  1
;  309     1031  2 BEGIN
;  310     1032  2
;  311     1033  2 LOCAL
;  312     1034  2     rte;
;  313     1035  2
;  314     1036  2
;  315     1037  2 write_command_q [0] = write_command_q [0];
;  316     1038  2 write_command_q [1] = write_command_q [0];
;  317     1039  2
;  318     1040  2 !
;  319     1041  2 ! Assign a channel to the port
;  320     1042  2 !
;  321   P 1043  2 Return_if_error ( LIB$ASN_WTH_MBX (.port_name, msgsiz, msgsiz,
;  322     1044  2                                   port_chan, port_mbx_chan) );
;  323     1045  2 !
;  324     1046  2 ! save away initial characteristics of the command terminal
;  325     1047  2 !
;  326     1048  2 Return_if_error (sense_mode (.command_chan, command_char));
;  327     1049  2 Return_if_error (sense_mode (.port_chan, port_char));
;  328     1050  2 !
;  329     1051  2 ! set delay delta
;  330     1052  2 !
;  331     1053  2 !delay_time [0] = -5000000;     ! 5 million is 1/2 second
;  332     1054  2 delay_time [0]  = -500000;      ! *** small time
```

```
;  333          1055  2 delay_time [1] = -1;
;  334          1056  2 !
;  335          1057  2 ! declare an exit handler here to reset command terminal characteristics
;  336          1058  2 !
;  337          1059  2 $DCLEXH (DESBLK = exit_block);
;  338          1060  2 !
;  339          1061  2 ! Set both the port line and the command line into passthru mode,
;  340          1062  2 ! this will enable everything except XON and XOFF.
;  341          1063  2 !
;  342          1064  2 Return_if_error ( set_passthru_mode (.port_chan, port_char, true));
;  343          1065  2 Return_if_error ( set_passthru_mode (.command_chan, command_char, false));
;  344          1066  2
;  345          1067  2 rte = allocate_buf (rte$c_length);              ! allocate
;  346          1068  2 read_port_mbx (.rte);
;  347          1069  2
;  348          1070  2 RETURN (ss$_normal);
;  349          1071  2
;  350          1072  1 END;                                           ! End of routine INITIALIZE


                                                        .EXTRN  SYS$DCLEXH

                                        0004 00000 INITIALIZE:
                                                        .WORD   Save R2                                 ; 0998
                            52  0000'  CF  9E 00002       MOVAB   WRITE_COMMAND_Q, R2
                            62          62  9E 00007       MOVAB   WRITE_COMMAND_Q, WRITE_COMMAND_Q    ; 1037
                        04  A2          62  9E 0000A       MOVAB   WRITE_COMMAND_Q, WRITE_COMMAND_Q+4  ; 1038
                              E0    A2  9F 0000E       PUSHAB  PORT_MBX_CHAN                           ; 1044
                              DE    A2  9F 00011       PUSHAB  PORT_CHAN
                              F4    A2  9F 00014       PUSHAB  MSGSIZ
                              F4    A2  9F 00017       PUSHAB  MSGSIZ
                              04    AC  DD 0001A       PUSHL   PORT_NAME
                00000000G     00        05  FB 0001D       CALLS   #5, LIB$ASN_WTH_MBX
                              69        50  E9 00024       BLBC    STATUS, 1$
                                    08  A2  9F 00027       PUSHAB  COMMAND_CHAR                        ; 1048
                              7E    DC  A2  3C 0002A       MOVZWL  COMMAND_CHAN, -(SP)
                          0000V     CF      02  FB 0002E       CALLS   #2, SENSE_MODE
                              5A        50  E9 00033       BLBC    STATUS, 1$
                                    14  A2  9F 00036       PUSHAB  PORT_CHAR                           ; 1049
                              7E    DE  A2  3C 00039       MOVZWL  PORT_CHAN, -(SP)
                          0000V     CF      02  FB 0003D       CALLS   #2, SENSE_MODE
                              4B        50  E9 00042       BLBC    STATUS, 1$
                            F8  A2  FFF85EE0  8F  D0 00045       MOVL    #-500000, DELAY_TIME        ; 1054
                            FC  A2          01  CE 0004D       MNEGL   #1, DELAY_TIME+4                ; 1055
                                    24  A2  9F 00051       PUSHAB  EXIT_BLOCK                          ; 1059
                00000000G     00        01  FB 00054       CALLS   #1, SYS$DCLEXH
                                        01  DD 0005B       PUSHL   #1                                  ; 1064
                                    14  A2  9F 0005D       PUSHAB  PORT_CHAR
                              7E    DE  A2  3C 00060       MOVZWL  PORT_CHAN, -(SP)
                          0000V     CF      03  FB 00064       CALLS   #3, SET_PASSTHRU_MODE
                              24        50  E9 00069       BLBC    STATUS, 1$
                                        7E  D4 0006C       CLRL    -(SP)                               ; 1065
                                    08  A2  9F 0006E       PUSHAB  COMMAND_CHAR
                              7E    DC  A2  3C 00071       MOVZWL  COMMAND_CHAN, -(SP)
                          0000V     CF      03  FB 00075       CALLS   #3, SET_PASSTHRU_MODE
                              13        50  E9 0007A       BLBC    STATUS, 1$
```

```
                7E      64    8F  9A 0007D          MOVZBL  #100, -(SP)                          ; 1067
        0000V   CF            01  FB 00081          CALLS   #1, ALLOCATE_BUF                    ;
                              50  DD 00086          PUSHL   R1                                  ; 1068
        0000V   CF            01  FB 00088          CALLS   #1, READ_PORT_MBX                   ;
                50            01  D0 0008D          MOVL    #1, R0                              ; 1070
                              04  00090 1$:         RET                                         ; 1072

; Routine Size:  145 bytes,    Routine Base:  $CODE$ + 003A

;  351          1073  1
```

```
;  353     1074  1 ROUTINE DIAL_NUMBER =
;  354     1075  1
;  355     1076  1 !++
;  356     1077  1 !
;  357     1078  1 ! FUNCTIONAL DESCRIPTION:
;  358     1079  1 !
;  359     1080  1 !       Look for DCL qualifier /DIAL=xyz-abcd and dial that number
;  360     1081  1 !       appropriately.
;  361     1082  1 !
;  362     1083  1 !
;  363     1084  1 ! CALLING SEQUENCE:
;  364     1085  1 !
;  365     1086  1 ! INPUT PARAMETERS:
;  366     1087  1 !       NONE
;  367     1088  1 !
;  368     1089  1 ! IMPLICIT INPUTS:
;  369     1090  1 !       NONE
;  370     1091  1 !
;  371     1092  1 ! OUTPUT PARAMETERS:
;  372     1093  1 !       NONE
;  373     1094  1 !
;  374     1095  1 ! IMPLICIT OUTPUTS:
;  375     1096  1 !       NONE
;  376     1097  1 !
;  377     1098  1 ! ROUTINE VALUE:
;  378     1099  1 !       NONE
;  379     1100  1 !
;  380     1101  1 ! SIDE EFFECTS:
;  381     1102  1 !       NONE
;  382     1103  1 !
;  383     1104  1 !--
;  384     1105  1
;  385     1106  2 BEGIN
;  386     1107  2
;  387     1108  2 LOCAL
;  388     1109  2     image_desc:         BBLOCK [dsc$c_s_bln],
;  389     1110  2     number_desc:        BBLOCK [dsc$c_s_bln],
;  390     1111  2     modem_desc:         BBLOCK [dsc$c_s_bln],
;  391     1112  2     file_desc:          BBLOCK [dsc$c_s_bln],
;  392     1113  2     dial_addr,
;  393     1114  2     status;
;  394     1115  2
;  395     1116  2 status = CLI$PRESENT ($DESCRIPTOR('DIAL'));      ! check for /DIAL=
;  396     1117  2 IF NOT .status THEN RETURN (SS$_NORMAL);         ! return if no /DIAL
;  397     1118  2
;  398     1119  2 CH$FILL (0, dsc$c_s_bln, number_desc);           ! zero desc.
;  399     1120  2 number_desc [dsc$b_class] = dsc$k_class_d;       ! set to dynamic allocation
;  400     1121  2 CH$MOVE (dsc$c_s_bln, number_desc, modem_desc); ! copy to other desc.
;  401     1122  2 CH$MOVE (dsc$c_s_bln, number_desc, image_desc); ! copy to other desc.
;  402     1123  2 !
;  403     1124  2 ! get NUMBER =
;  404     1125  2 !
;  405     1126  2 return_if_error (CLI$GET_VALUE ($DESCRIPTOR ('NUMBER'), number_desc));
;  406     1127  2 !**lib$put_output (number_desc);
;  407     1128  2 !
;  408     1129  2 ! get MODEM_TYPE =
;  409     1130  2 !
```

```
;  410          1131  2 status = CLI$GET_VALUE ($DESCRIPTOR ('MODEM_TYPE'), modem_desc);
;  411          1132  2 IF .status EQL CLI$_ABSENT THEN
;  412          1133  2     BEGIN
;  413          1134  2     !
;  414          1135  2     ! CLI parse can't return default if MODEM_TYPE not specified,
;  415          1136  2     ! defaulting is done by hand.
;  416          1137  2     !
;  417          1138  2     modem_desc [dsc$w_length]  = %CHARCOUNT ('DF03');
;  418          1139  2     modem_desc [dsc$a_pointer] = UPLIT BYTE ('DF03');
;  419          1140  2     END;
;  420          1141  2 !**lib$put_output (modem_desc);
;  421          1142  2
;  422        P 1143  2 return_if_error (
;  423          1144  2     STR$CONCAT (image_desc, $DESCRIPTOR ('DTE_'), modem_desc));
;  424          1145  2
;  425        P 1146  2 return_if_error (
;  426          1147  2     LIB$FIND_IMAGE_SYMBOL (image_desc, $DESCRIPTOR ('DIAL_ROUTINE'),dial_addr));
;  427          1148  2 !
;  428          1149  2 ! Call the newly activated image...
;  429          1150  2 !
;  430          1151  2 status = (.dial_addr)(number_desc, .port_chan, .command_chan);
;  431          1152  2
;  432          1153  3 RETURN (.status)
;  433          1154  1 END;                                    ! End of routine


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    4C  41  49  44  0000C P.AAD:  .ASCII  \DIAL\                          ;
                                          00000004  00010 P.AAC:  .LONG   4                               ;
                                          00000000' 00014         .ADDRESS P.AAD                        ;
                                52  45  42  4D  55  4E  00018 P.AAF:  .ASCII  \NUMBER\                    ;
                                                00001E          .BLKB   2                               ;
                                          00000006  00020 P.AAE:  .LONG   6                               ;
                                          00000000' 00024         .ADDRESS P.AAF                        ;
                    45  50  59  54  5F  4D  45  44  4F  4D  00028 P.AAH:  .ASCII  \MODEM_TYPE\            ;
                                                00032           .BLKB   2                               ;
                                          0000000A  00034 P.AAG:  .LONG   10                              ;
                                          00000000' 00038         .ADDRESS P.AAH                        ;
                                    33  30  46  44  0003C P.AAI:  .ASCII  \DF03\                        ;
                                    5F  45  54  44  00040 P.AAK:  .ASCII  \DTE_\                        ;
                                          00000004  00044 P.AAJ:  .LONG   4                               ;
                                          00000000' 00048         .ADDRESS P.AAK                        ;
45  4E  49  54  55  4F  52  5F  4C  41  49  44  0004C P.AAM:  .ASCII  \DIAL_ROUTINE\                     ;
                                          0000000C  00058 P.AAL:  .LONG   12                              ;
                                          00000000' 0005C         .ADDRESS P.AAM                        ;


                                                        .PSECT  $CODE$,NOWRT,2

                                                01FC 00000 DIAL_NUMBER:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8            ; 1074
                                58 00000000G    00  9E 00002           MOVAB   CLI$GET_VALUE, R8        ;
                                57      0000'   CF  9E 00009           MOVAB   P.AAC, R7                ;
                                5E              24  C2 0000E           SUBL2   #36, SP                  ;
```

```
                                    57  DD 00011       PUSHL   R7                                  ; 1116
                 00000000G  00      01  FB 00013       CALLS   #1, CLI$PRESENT                     ;
                            56      50  D0 0001A       MOVL    R0, STATUS                          ;
                            04      56  E8 0001D       BLBS    STATUS, 1$                          ; 1117
                            50      01  D0 00020       MOVL    #1, R0                              ;
                                    04     00023       RET                                         ;
      08            00      6E      00  2C 00024 1$:   MOVC5   #0, (SP), #0, #8, NUMBER_DESC       ; 1119
                                14  AE     00029                                               ;
                      17    AE      02  90 0002B       MOVB    #2, NUMBER_DESC+3                   ; 1120
             0C    AE 14    AE      08  28 0002F       MOVC3   #8, NUMBER_DESC, MODEM_DESC         ; 1121
             1C    AE 14    AE      08  28 00035       MOVC3   #8, NUMBER_DESC, IMAGE_DESC         ; 1122
                                14  AE  9F 0003B       PUSHAB  NUMBER_DESC                         ; 1126
                                10  A7  9F 0003E       PUSHAB  P.AAE                               ;
                            68      02  FB 00041       CALLS   #2, CLI$GET_VALUE                   ;
                            57      50  E9 00044       BLBC    STATUS, 3$                          ;
                                0C  AE  9F 00047       PUSHAB  MODEM_DESC                          ; 1131
                                24  A7  9F 0004A       PUSHAB  P.AAG                               ;
                            68      02  FB 0004D       CALLS   #2, CLI$GET_VALUE                   ;
                            56      50  D0 00050       MOVL    R0, STATUS                          ;
                 00000000G  8F      56  D1 00053       CMPL    STATUS, #CLI$_ABSENT                ; 1132
                                    09  12 0005A       BNEQ    2$                                  ;
                      0C    AE      04  B0 0005C       MOVW    #4, MODEM_DESC                      ; 1138
                      10    AE  2C  A7  9E 00060       MOVAB   P.AAI, MODEM_DESC+4                 ; 1139
                                0C  AE  9F 00065 2$:   PUSHAB  MODEM_DESC                          ; 1144
                                34  A7  9F 00068       PUSHAB  P.AAJ                               ;
                                24  AE  9F 0006B       PUSHAB  IMAGE_DESC                          ;
                 00000000G  00      03  FB 0006E       CALLS   #3, STR$CONCAT                      ;
                            26      50  E9 00075       BLBC    STATUS, 3$                          ;
                                    5E  DD 00078       PUSHL   SP                                  ; 1147
                                48  A7  9F 0007A       PUSHAB  P.AAL                               ;
                                24  AE  9F 0007D       PUSHAB  IMAGE_DESC                          ;
                 00000000G  00      03  FB 00080       CALLS   #3, LIB$FIND_IMAGE_SYMBOL           ;
                            14      50  E9 00087       BLBC    STATUS, 3$                          ;
                            7E   0000'  CF  3C 0008A   MOVZWL  COMMAND_CHAN, -(SP)                 ; 1151
                            7E   0000'  CF  3C 0008F   MOVZWL  PORT_CHAN, -(SP)                    ;
                                1C  AE  9F 00094       PUSHAB  NUMBER_DESC                         ;
                      0C    BE      03  FB 00097       CALLS   #3, @DIAL_ADDR                      ;
                            56      50  D0 0009B       MOVL    R0, STATUS                          ;
                                    04     0009E 3$:   RET                                         ; 1154

; Routine Size:  159 bytes,    Routine Base:  $CODE$ + 00CB


;  434          1155  1
;  435          1156  1
```

```
;  437          1157  1 ROUTINE TRANSMIT =
;  438          1158  1
;  439          1159  1 !++
;  440          1160  1 !
;  441          1161  1 ! FUNCTIONAL DESCRIPTION:
;  442          1162  1 !
;  443          1163  1 !       Set up read's and AST and return.  Entire program is AST driven
;  444          1164  1 !       after the return from this routine.
;  445          1165  1 !
;  446          1166  1 ! CALLING SEQUENCE:
;  447          1167  1 !
;  448          1168  1 !       TRANSMIT();
;  449          1169  1 !
;  450          1170  1 ! INPUT PARAMETERS:
;  451          1171  1 !       NONE
;  452          1172  1 !
;  453          1173  1 ! IMPLICIT INPUTS:
;  454          1174  1 !       NONE
;  455          1175  1 !
;  456          1176  1 ! OUTPUT PARAMETERS:
;  457          1177  1 !       NONE
;  458          1178  1 !
;  459          1179  1 ! IMPLICIT OUTPUTS:
;  460          1180  1 !       NONE
;  461          1181  1 !
;  462          1182  1 ! ROUTINE VALUE:
;  463          1183  1 !       NONE
;  464          1184  1 !
;  465          1185  1 ! SIDE EFFECTS:
;  466          1186  1 !       NONE
;  467          1187  1 !
;  468          1188  1 !--
;  469          1189  1
;  470          1190  2 BEGIN
;  471          1191  2 LOCAL
;  472          1192  2     buffer_cr:  BYTE INITIAL (13),
;  473          1193  2     rte:        REF $BBLOCK;                    ! rte data block
;  474          1194  2
;  475          1195  2 IF .INDFLAG THEN
;  476          1196  2     !
;  477          1197  2     ! Set up for initial RMS reads.
;  478          1198  2     !
;  479          1199  3     Return_if_error (Read_sysinput ())
;  480          1200  2 ELSE
;  481          1201  3     BEGIN
;  482          1202  3     !
;  483          1203  3     ! Set up dual reads on command channel
;  484          1204  3     !
;  485          1205  3     rte = allocate_buf (rte$c_length);          ! allocate
;  486          1206  3     Read_command_kb (.rte);
;  487          1207  3     rte = allocate_buf (rte$c_length);          ! allocate
;  488          1208  3     Read_command_kb (.rte);
;  489          1209  2     END;
;  490          1210  2 !
;  491          1211  2 ! Do a read timeout on the port to set up reading
;  492          1212  2 !
;  493          1213  2 Read_port_timed (1);
```

```
;  494          1214  2
;  495          1215  2 !
;  496          1216  2 ! Kick the whole thing off by sending a CR down the line.  If it's a
;  497          1217  2 ! VAX down there, LOGIN should start up.
;  498          1218  2 !
;  499        P 1219  2 quit_if_error (
;  500        P 1220  2     $QIO (CHAN  = .port_chan,
;  501        P 1221  2           FUNC  = io$_writevblk,
;  502        P 1222  2           P1    = buffer_cr,
;  503        P 1223  2           P2    = 1,
;  504        P 1224  2           P4    = 0)                    ! 0 forces no carriage control
;  505          1225  2           );
;  506          1226  2
;  507          1227  2 RETURN (ss$_normal);
;  508          1228  2
;  509          1229  2
;  510          1230  1 END;                                    ! End of routine


                                                        .EXTRN  SYS$QIO, SYS$WAKE

                                    0004 00000 TRANSMIT:
                                                        .WORD   Save R2                         ; 1157
                              5E            04  C2 00002        SUBL2   #4, SP                  ;
                              6E            0D  90 00005        MOVB    #13, BUFFER_CR          ; 1190
                              09 00000000G  00  E9 00008        BLBC    INDFLAG, 1$             ; 1195
                        0000V CF            00  FB 0000F        CALLS   #0, READ_SYSINPUT       ; 1199
                              27            50  E8 00014        BLBS    STATUS, 2$              ;
                                            04  00017        RET                             ;
                              7E      64    8F  9A 00018 1$:    MOVZBL  #100, -(SP)             ; 1205
                        0000V CF            01  FB 0001C        CALLS   #1, ALLOCATE_BUF        ;
                              52            50  D0 00021        MOVL    R0, RTE                 ;
                                            52  DD 00024        PUSHL   RTE                     ; 1206
                        0000V CF            01  FB 00026        CALLS   #1, READ_COMMAND_KB     ;
                              7E      64    8F  9A 0002B        MOVZBL  #100, -(SP)             ; 1207
                        0000V CF            01  FB 0002F        CALLS   #1, ALLOCATE_BUF        ;
                              52            50  D0 00034        MOVL    R0, RTE                 ;
                                            52  DD 00037        PUSHL   RTE                     ; 1208
                        0000V CF            01  FB 00039        CALLS   #1, READ_COMMAND_KB     ;
                                            01  DD 0003E 2$:    PUSHL   #1                      ; 1213
                        0000V CF            01  FB 00040        CALLS   #1, READ_PORT_TIMED     ;
                                            7E  7C 00045        CLRQ    -(SP)                   ; 1225
                                            7E  7C 00047        CLRQ    -(SP)                   ;
                                            01  DD 00049        PUSHL   #1                      ;
                                      14    AE  9F 0004B        PUSHAB  BUFFER_CR               ;
                                            7E  7C 0004E        CLRQ    -(SP)                   ;
                                      7E    30  7D 00050        MOVQ    #48, -(SP)              ;
                                      7E  0000' CF  3C 00053    MOVZWL  PORT_CHAN, -(SP)        ;
                                            7E  D4 00058        CLRL    -(SP)                   ;
                  00000000G             00  FB 0005A        CALLS   #12, SYS$QIO            ;
                                        52  50  D0 00061        MOVL    R0, STATUS              ;
                                        1B  52  E8 00064        BLBS    STATUS, 3$              ;
                  00000000G             00  01  D0 00067        MOVL    #1, WAKEFLAG            ;
                                            7E  7C 0006E        CLRQ    -(SP)                   ;
                  00000000G             00  02  FB 00070        CALLS   #2, SYS$WAKE            ;
                  00000000G             00  52  D0 00077        MOVL    STATUS, RETSTATUS       ;
```

```
                        50              52  D0 0007E           MOVL    STATUS, R0                              ;
                                            04 00081           RET                                             ;
                        50              01  D0 00082 3$:       MOVL    #1, R0                                  ; 1227
                                            04 00085           RET                                             ; 1230

; Routine Size:  134 bytes,    Routine Base:  $CODE$ + 016A

;  511          1231  1
```

```
;   513          1232  1 ROUTINE Read_command_kt (rte: REF $BBLOCK) : NOVALUE =
;   514          1233  1
;   515          1234  1 !++
;   516          1235  1 !
;   517          1236  1 ! FUNCTIONAL DESCRIPTION:
;   518          1237  1 !
;   519          1238  1 !       Read from the command keyboard a character at a time.
;   520          1239  1 !
;   521          1240  1 ! CALLING SEQUENCE:
;   522          1241  1 !
;   523          1242  1 !       Read_command_kb ();
;   524          1243  1 !
;   525          1244  1 ! INPUT PARAMETERS:
;   526          1245  1 !       NONE
;   527          1246  1 !
;   528          1247  1 ! IMPLICIT INPUTS:
;   529          1248  1 !
;   530          1249  1 !       COMMAND_CHAN
;   531          1250  1 !
;   532          1251  1 ! OUTPUT PARAMETERS:
;   533          1252  1 !       NONE
;   534          1253  1 !
;   535          1254  1 ! IMPLICIT OUTPUTS:
;   536          1255  1 !       NONE
;   537          1256  1 !
;   538          1257  1 ! ROUTINE VALUE:
;   539          1258  1 !
;   540          1259  1 !       status
;   541          1260  1 !
;   542          1261  1 ! SIDE EFFECTS:
;   543          1262  1 !
;   544          1263  1 !       A QIO to SYS$COMMAND is done.
;   545          1264  1 !
;   546          1265  1 !--
;   547          1266  1
;   548          1267  2 BEGIN
;   549          1268  2
;   550          1269  2 !
;   551          1270  2 ! Queue a request for a single character
;   552          1271  2 !
;   553        P 1272  2 Quit_if_error (
;   554        P 1273  2     $QIO (CHAN  = .command_chan,
;   555        P 1274  2           FUNC  = io$_readvblk OR io$m_noecho,
;   556        P 1275  2           ASTADR  = Read_kb_ast,
;   557        P 1276  2           ASTPRM  = .rte,
;   558        P 1277  2           IOSB  = rte [rte$q_iosb],
;   559        P 1278  2           P1  = rte [rte$t_buf],
;   560        P 1279  2           P2  = 1)
;   561          1280  2           );
;   562          1281  2
;   563          1282  1 END;                                    ! End of routine Read_command_kb


                                                0004 00000 READ_COMMAND_KB:
```

```
                                                        .WORD   Save R2                          ; 1232
                                   7E  7C 00002         CLRQ    -(SP)                            ; 1280
                                   7E  7C 00004         CLRQ    -(SP)                            ;
                                   01  DD 00006         PUSHL   #1                               ;
        7E        04     AC        14  C1 00008         ADDL3   #20, RTE, -(SP)                  ;
                          04       AC  DD 0000D         PUSHL   RTE                              ;
                       0000V       CF  9F 00010         PUSHAB  READ_KB_AST                      ;
        7E        04     AC        0C  C1 00014         ADDL3   #12, RTE, -(SP)                  ;
                         7E     71 8F  9A 00019         MOVZBL  #113, -(SP)                      ;
                         7E  0000' CF  3C 0001D         MOVZWL  COMMAND_CHAN, -(SP)              ;
                                   7E  D4 00022         CLRL    -(SP)                            ;
             00000000G   00        0C  FB 00024         CALLS   #12, SYS$QIO                     ;
                         52        50  D0 0002B         MOVL    R0, STATUS                       ;
                         17        52  E8 0002E         BLBS    STATUS, 1$                       ;
             00000000G   00        01  D0 00031         MOVL    #1, WAKEFLAG                     ;
                                   7E  7C 00038         CLRQ    -(SP)                            ;
             00000000G   00        02  FB 0003A         CALLS   #2, SYS$WAKE                     ;
             00000000G   00        52  D0 00041         MOVL    STATUS, RETSTATUS                ;
                                   04 00048 1$:         RET                                      ; 1282

; Routine Size:  73 bytes,    Routine Base:  $CODE$ + 01F0
```

```
;  565          1283  1 ROUTINE read_kb_ast (rte: REF $BBLOCK) : NOVALUE =
;  566          1284  1
;  567          1285  1 !++
;  568          1286  1 !
;  569          1287  1 ! FUNCTIONAL DESCRIPTION:
;  570          1288  1 !
;  571          1289  1 !
;  572          1290  1 !
;  573          1291  1 ! CALLING SEQUENCE:
;  574          1292  1 !
;  575          1293  1 ! INPUT PARAMETERS:
;  576          1294  1 !       NONE
;  577          1295  1 !
;  578          1296  1 ! IMPLICIT INPUTS:
;  579          1297  1 !       NONE
;  580          1298  1 !
;  581          1299  1 ! OUTPUT PARAMETERS:
;  582          1300  1 !       NONE
;  583          1301  1 !
;  584          1302  1 ! IMPLICIT OUTPUTS:
;  585          1303  1 !       NONE
;  586          1304  1 !
;  587          1305  1 ! ROUTINE VALUE:
;  588          1306  1 !       NONE
;  589          1307  1 !
;  590          1308  1 ! SIDE EFFECTS:
;  591          1309  1 !       NONE
;  592          1310  1 !
;  593          1311  1 !--
;  594          1312  1
;  595          1313  2 BEGIN
;  596          1314  2
;  597          1315  2 BIND
;  598          1316  2     IOSB_WORD = rte [rte$q_iosb] : VECTOR [,WORD],
;  599          1317  2     IOSB_BYTE = rte [rte$q_iosb] : VECTOR [,BYTE],
;  600          1318  2     char = rte [rte$t_buf] : BYTE;
;  601          1319  2
;  602          1320  2 LOCAL
;  603          1321  2     outdesc:    VECTOR [2],
;  604          1322  2     outbuf:     VECTOR [132,BYTE];
;  605          1323  2 !
;  606          1324  2 ! %% report stats
;  607          1325  2 !
;  608          1326  2 outdesc [1] = outbuf;
;  609          1327  2
;  610          1328  2 IF .char EQL 28 THEN ! *** look for ^\
;  611          1329  3     BEGIN
;  612          1330  3     ! PRINT (%ASCID 'write_active_count:!_!UL', .write_active_count);
;  613          1331  3     ! PRINT (%ASCID 'port_count!_!UL',          .port_count);
;  614          1332  3     ! PRINT (%ASCID 'read_pendi ,:!_!UL',       .read_pending);
;  615          1333  3     ! dump_stats ();
;  616          1334  3     !debug_ast ();
;  617          1335  3
;  618          1336  3     quit (ss$_normal);
;  619          1337  2     END;
;  620          1338  2
;  621          1339  2 iosb_word [w_riosb_datasize] = .iosb_word [w_riosb_datasize] +
```

```
;  622          1340  2                                   .iosb_byte [b_riosb_termsize];
;  623          1341  2 !
;  624          1342  2 ! adjust for TTDRIVER bug
;  625          1343  2 !
;  626          1344  2 IF .iosb_byte [b_riosb_termchar] EQL 0 THEN
;  627          1345  2 iosb_word [w_riosb_datasize] = .iosb_word [w_riosb_datasize] -
;  628          1346  2                                   .iosb_byte [b_riosb_termsize];
;  629          1347  2
;  630          1348  2 IF .IOSB_WORD [w_riosb_datasize] NEQ 0 THEN     ! if there is data in the buffer
;  631        P 1349  2     quit_if_error (
;  632        P 1350  2         $QIO (CHAN      = .port_chan,
;  633        P 1351  2               FUNC      = io$_writevblk,
;  634        P 1352  2               ASTADR    = read_command_kb,
;  635        P 1353  2               ASTPRM    = .rte,
;  636        P 1354  2               IOSB      = rte [rte$q_iosb],
;  637        P 1355  2               P1        = rte [rte$t_buf],
;  638        P 1356  2               P2        = .iosb_word [w_riosb_datasize],
;  639        P 1357  2               P4        = 0)                      ! 0 forces no carriage control
;  640          1358  2               );
;  641          1359  2
;  642          1360  1 END;                                  ! End of routine read_kb_ast



                                007C 00000 READ_KB_AST:
                                           .WORD   Save R2,R3,R4,R5,R6                                  ; 1283
                            56 00000000G   00  9E 00002    MOVAB   WAKEFLAG, R6                        ;
                            55 00000000G   00  9E 00009    MOVAB   RETSTATUS, R5                       ;
                            54 00000000G   00  9E 00010    MOVAB   SYS$WAKE, R4                        ;
                            5E      FF74   CE  9E 00017    MOVAB   -140(SP), SP                        ;
                        52  04      AC         0C  C1 0001C    ADDL3   #12, RTE, R2                     ; 1316
                        53  04      AC         14  C1 00021    ADDL3   #20, RTE, R3                     ; 1318
                            FC      AD         6E  9E 00026    MOVAB   OUTBUF, OUTDESC+4                ; 1326
                            1C                 63  91 0002A    CMPB    (R3), #28                        ; 1328
                                               0C  12 0002D    BNEQ    1$                               ;
                            66                 01  D0 0002F    MOVL    #1, WAKEFLAG                     ; 1336
                                               7E  7C 00032    CLRQ    -(SP)                            ;
                            64                 02  FB 00034    CALLS   #2, SYS$WAKE                     ;
                            65                 01  D0 00037    MOVL    #1, RETSTATUS                    ;
                                               04  00 0003A    RET                                      ;
                            50          06     A2  9A 0003B 1$:    MOVZBL  6(R2), R0                    ; 1340
                        02  A2                 50  A0 0003F    ADDW2   R0, 2(R2)                        ;
                                    04     A2  95  00043    TSTB    4(R2)                            ; 1344
                                               08  12 00046    BNEQ    2$                               ;
                            50          06     A2  9A 00048    MOVZBL  6(R2), R0                    ; 1346
                        02  A2                 50  A2 0004C    SUBW2   R0, 2(R2)                        ;
                                    02     A2  B5  00050 2$:    TSTW    2(R2)                            ; 1348
                                               34  13 00053    BEQL    3$                               ;
                                               7E  7C 00055    CLRQ    -(SP)                            ; 1358
                                               7E  7C 00057    CLRQ    -(SP)                            ;
                            7E          02     A2  3C 00059    MOVZWL  2(R2), -(SP)                     ;
                                               53  DD 0005D    PUSHL   R3                               ;
                                        04     AC  DD 0005F    PUSHL   RTE                              ;
                                    FF51       CF  9F 00062    PUSHAB  READ_COMMAND_KB                  ;
                                               52  DD 00066    PUSHL   R2                               ;
```

```
                                        30  DD 00068          PUSHL   #48
                              7E        CF  3C 0006A          MOVZWL  PORT_CHAN, -(SP)
                                        7E  D4 0006F          CLRL    -(SP)
                    00000000G 00        0C  FB 00071          CALLS   #12, SYS$QIO
                              52        50  D0 00078          MOVL    R0, STATUS
                              0B        52  E8 0007B          BLBS    STATUS, 3$
                              66        01  D0 0007E          MOVL    #1, WAKEFLAG
                                        7E  7C 00081          CLRQ    -(SP)
                              64        02  FB 00083          CALLS   #2, SYS$WAKE
                              65        52  D0 00086          MOVL    STATUS, RETSTATUS
                                        04  00089 3$:        RET                                     ; 1360

; Routine Size:  138 bytes,    Routine Base:  $CODE$ + 0239
```

```
;  644          1361  1 ROUTINE Read_port_timed (nodelay) : NOVALUE =
;  645          1362  1
;  646          1363  1 !++
;  647          1364  1 !
;  648          1365  1 ! FUNCTIONAL DESCRIPTION:
;  649          1366  1 !
;  650          1367  1 !       Start reading from port by doing a read with a generous buffer
;  651          1368  1 !       and a 0 second timeout.  The effect of this is to read typeahead
;  652          1369  1 !       and return (to AST) immediately.
;  653          1370  1 !
;  654          1371  1 ! CALLING SEQUENCE:
;  655          1372  1 !
;  656          1373  1 !       Read_port_timed ();
;  657          1374  1 !                                                                              ;
;  658          1375  1 ! INPUT PARAMETERS:
;  659          1376  1 !
;  660          1377  1 !       nodelay - 0 if no hurry                                                ;
;  661          1378  1 !                 1 if urgent request
;  662          1379  1 !
;  663          1380  1 ! IMPLICIT INPUTS:
;  664          1381  1 !
;  665          1382  1 !       PORT_CHAN
;  666          1383  1 !
;  667          1384  1 ! OUTPUT PARAMETERS:
;  668          1385  1 !       NONE
;  669          1386  1 !
;  670          1387  1 ! IMPLICIT OUTPUTS:
;  671          1388  1 !       NONE
;  672          1389  1 !
;  673          1390  1 ! ROUTINE VALUE:
;  674          1391  1 !       Status
;  675          1392  1 !
;  676          1393  1 ! SIDE EFFECTS:
;  677          1394  1 !       NONE
;  678          1395  1 !
;  679          1396  1 !--
;  680          1397  1
;  681          1398  2 BEGIN
;  682          1399  2
;  683          1400  2 LOCAL
;  684          1401  2     rte:        REF $BBLOCK;            ! rte data block
;  685          1402  2
;  686          1403  2 IF .nodelay THEN
;  687          1404  3     BEGIN
;  688          1405  3     cnt_rp_timed = .cnt_rp_timed + 1;           ! %% counter
;  689          1406  3     rte = allocate_buf (rte$c_length);          ! allocate
;  690          1407  3     port_count = .port_count + 1;                       ! increment outstanding read
;  691          1408  3     IF .port_count GTR .max_port_count THEN max_port_count = .port_count; !%%
;  692          1409  3     read_pending = false;                               ! clear read pending
;  693          1410  3     !
;  694          1411  3     ! Read with a zero second time out from the port channel. The effect of this
;  695          1412  3     ! is to read the buffer and return immediately, with or without a terminator.
;  696          1413  3     !
;  697        P 1414  3     quit_if_error (
;  698        P 1415  3         $QIO (CHAN = .port_chan,
;  699        P 1416  3             FUNC   = io$_readvblk OR io$m_timed OR io$m_noecho,
;  700        P 1417  3             ASTADR = Read_port_ast,
```

```
;  701          P 1418  3             ASTPRM = .rte,
;  702          P 1419  3             IOSB   = rte [rte$q_iosb],
;  703          P 1420  3             P1     = rte [rte$t_buf],
;  704          P 1421  3             P2     = rte$c_buflen)
;  705            1422  3         );
;  706            1423  3     END
;  707            1424  2 ELSE
;  708            1425  3     BEGIN
;  709            1426  3     IF .port_count GEQ rte$c_portread_max THEN RETURN;
;  710            1427  3     IF NOT .timer_pending THEN
;  711            1428  4         BEGIN
;  712            1429  4         timer_pending = true;
;  713          P 1430  4         quit_if_error (
;  714          P 1431  4             $SETIMR (DAYTIM = delay_time,
;  715          P 1432  4                 ASTADR = read_delay_ast)
;  716            1433  4                 );
;  717            1434  3         END;
;  718            1435  2     END;
;  719            1436  1 END;                                   ! End of routine Read_port_timed


                                000C 00000 READ_PORT_TIMED:
                                           .WORD   Save R2,R3                          ; 1361
                     53   0000'  CF  9E 00002    MOVAB   PORT_COUNT, R3              ;
                     45     04   AC  E9 00007    BLBC    NODELAY, 2$                 ; 1403
                           FDD8   C3  D6 0000B    INCL    CNT_RP_TIMED                ; 1405
                     7E     64   8F  9A 0000F    MOVZBL  #100, -(SP)                 ; 1406
               0000V CF           01  FB 00013    CALLS   #1, ALLOCATE_BUF            ;
                                  63  D6 00018    INCL    PORT_COUNT                  ; 1407
                FDD4 C3           63  D1 0001A    CMPL    PORT_COUNT, MAX_PORT_COUNT  ; 1408
                                  05  15 0001F    BLEQ    1$                          ;
                FDD4 C3           63  D0 00021    MOVL    PORT_COUNT, MAX_PORT_COUNT  ;
                       0C   A3  D4 00026 1$:      CLRL    READ_PENDING                ; 1409
                     7E     7C  00029            CLRQ    -(SP)                       ; 1422
                     7E     7C  0002B            CLRQ    -(SP)
                     7E     50  8F  9A 0002D     MOVZBL  #80, -(SP)
                     14     A0  9F  00031        PUSHAB  20(RTE)
                             50  DD 00034        PUSHL   RTE
                          0000V  CF  9F 00036    PUSHAB  READ_PORT_AST
                       0C   A0  9F  0003A        PUSHAB  12(RTE)
                     7E     F1  8F  9A 0003D     MOVZBL  #241, -(SP)
                     7E     FA  A3  3C 00041     MOVZWL  PORT_CHAN, -(SP)
                                7E  D4 00045     CLRL    -(SP)
          00000000G  00           0C  FB 00047    CALLS   #12, SYS$QIO
                                  1F  11 0004E    BRB     3$
                     02           63  D1 00050 2$:      CMPL    PORT_COUNT, #2      ; 1426
                                  37  18 00053    BGEQ    4$
                     33     08   A3  E8 00055    BLBS    TIMER_PENDING, 4$           ; 1427
                08   A3           01  D0 00059    MOVL    #1, TIMER_PENDING           ; 1429
                                7E  D4 0005D     CLRL    -(SP)                       ; 1433
                          0000V  CF  9F 0005F    PUSHAB  READ_DELAY_AST
                       14   A3  9F  00063        PUSHAB  DELAY_TIME
                                7E  D4 00066     CLRL    -(SP)
          00000000G  00           04  FB 00068    CALLS   #4, SYS$SETIMR
```

```
                        52              50  D0 0006F 3$:    MOVL    R0, STATUS                              ;
                        17              52  E8 00072       BLBS    STATUS, 4$                              ;
                00000000G  00           01  D0 00075       MOVL    #1, WAKEFLAG                            ;
                                        7E  7C 0007C       CLRQ    -(SP)                                   ;
                00000000G  00           02  FB 0007E       CALLS   #2, SYS$WAKE                            ;
                00000000G  00           52  D0 00085       MOVL    STATUS, RETSTATUS                       ;
                                            04 0008C 4$:    RET                                             ; 1436

; Routine Size:  141 bytes,    Routine Base:  $CODE$ + 02C3
```

```
;  721          1437  1 ROUTINE Read_port_ast (rte: REF $BBLOCK) : NOVALUE =
;  722          1438  1
;  723          1439  1 !++
;  724          1440  1 !
;  725          1441  1 ! FUNCTIONAL DESCRIPTION.
;  726          1442  1 !
;  727          1443  1 !       Handle AST delivered from reading port line to host system.
;  728          1444  1 !       This data must be displayed on command port with all due
;  729          1445  1 !       dispatch.
;  730          1446  1 !
;  731          1447  1 !       If QIO did not finish with status of SS$_TIMEOUT, then there is
;  732          1448  1 !       still more data in the buffer, another timed read must be done.
;  733          1449  1 !
;  734          1450  1 ! CALLING SEQUENCE:
;  735          1451  1 !
;  736          1452  1 !       AST routine
;  737          1453  1 !
;  738          1454  1 ! INPUT PARAMETERS:
;  739          1455  1 !
;  740          1456  1 !       rte - buffer with QIO status and data
;  741          1457  1 !
;  742          1458  1 ! IMPLICIT INPUTS:
;  743          1459  1 !       NONE
;  744          1460  1 !
;  745          1461  1 ! OUTPUT PARAMETERS:
;  746          1462  1 !       NONE
;  747          1463  1 !
;  748          1464  1 ! IMPLICIT OUTPUTS:
;  749          1465  1 !       NONE
;  750          1466  1 !
;  751          1467  1 ! ROUTINE VALUE:
;  752          1468  1 !       NONE
;  753          1469  1 !
;  754          1470  1 ! SIDE EFFECTS:
;  755          1471  1 !       NONE
;  756          1472  1 !
;  757          1473  1 !--
;  758          1474  1
;  759          1475  2 BEGIN
;  760          1476  2 !
;  761          1477  2 ! Set up references to IOSB
;  762          1478  2 !
;  763          1479  2 BIND
;  764          1480  2     IOSB_WORD = rte [rte$q_iosb] : VECTOR [,WORD],
;  765          1481  2     IOSB_BYTE = rte [rte$q_iosb] : VECTOR [,BYTE];
;  766          1482  2
;  767          1483  2 IF  (.iosb_word [w_riosb_status] NEQ ss$_timeout) THEN
;  768          1484  3     BEGIN
;  769          1485  3     read_port_timed (0);
;  770          1486  3     cnt_rp_ast = .cnt_rp_ast + 1;
;  771          1487  2     END;
;  772          1488  2
;  773          1489  2 IF (NOT .iosb_word [w_riosb_status]) AND (.iosb_word [w_riosb_status] NEQ ss$_timeout) THEN
;  774          1490  2     quit (.iosb_word [w_riosb_status]);
;  775          1491  2
;  776          1492  2 iosb_word [w_riosb_datasize] = .iosb_word [w_riosb_datasize] +
;  777          1493  2                                 .iosb_byte [b_riosb_termsize];
```

```
;  778          1494  2 !
;  779          1495  2 ! adjust for TTDRIVER bug
;  780          1496  2 !
;  781          1497  2 IF .iosb_byte [b_riosb_termchar] EQL 0 THEN
;  782          1498  2 iosb_word [w_riosb_datasize] = .iosb_word [w_riosb_datasize] -
;  783          1499  2                                 .iosb_byte [b_riosb_termsize];
;  784          1500  2
;  785          1501  2 port_count = .port_count - 1;
;  786          1502  2 IF .iosb_word [w_riosb_datasize] EQL 0 THEN
;  787          1503  3     BEGIN                                                                  ;
;  788          1504  3     !
;  789          1505  3     ! don't bother with this guy
;  790          1506  3     !                                                                      ;
;  791          1507  3     cnt_rp_zero = .cnt_rp_zero + 1;      ! %% counter
;  792          1508  3     deallocate_buf (.rte);
;  793          1509  3 !***     port_count = .port_count - 1;
;  794          1510  3     RETURN;
;  795          1511  2     END;
;  796          1512  2
;  797          1513  2 cnt_rp_char = .cnt_rp_char + .iosb_word [w_riosb_datasize];  ! %% counter
;  798          1514  2
;  799          1515  2 INSQUE (.rte, .write_command_q [1]);    ! insert at end of queue
;  800          1516  2
;  801          1517  2 write_command ();                       ! check for start write
;  802          1518  2
;  803          1519  1 END;                                    ! End of routine Read_port_ast


                                000C 00000 READ_PORT_AST:
                                           .WORD   Save R2,R3                              ;  1437
52      04   AC                 0C   C1 00002           ADDL3   #12, RTE, R2                ;  1480
                                53   D4 00007           CLRL    R3                          ;  1483
        022C 8F                 62   B1 00009           CMPW    (R2), #556                  ;
                                0D   13 0000E           BEQL    1$                          ;
                                53   D6 00010           INCL    R3                          ;
                                7E   D4 00012           CLRL    -(SP)                       ;  1485
        FF5A CF                 01   FB 00014           CALLS   #1, READ_PORT_TIMED         ;
                          0000' CF   D6 00019           INCL    CNT_RP_AST                  ;  1486
                             1B   62   E8 0001D 1$:      BLBS    (R2), 2$                    ;  1489
                             18   53   E9 00020           BLBC    R3, 2$                      ;
    00000000G                00   01   D0 00023           MOVL    #1, WAKEFLAG                ;  1490
                                7E   7C 00026           CLRQ    -(SP)                       ;
    00000000G                00   02   FB 00028           CALLS   #2, SYS$WAKE                ;
    00000000G                00   62   3C 0002F           MOVZWL  (R2), RETSTATUS             ;
                                04   04 00032           RET                                 ;
                             50   06   A2   9A 00033 2$:  MOVZBL  6(R2), R0                  ;  1493
            02               A2        50   A0 00037     ADDW2   R0, 2(R2)                   ;
                                04   A2   95 0003B       TSTB    4(R2)                       ;  1497
                                08   12   00040         BNEQ    3$                          ;
                             50   06   A2   9A 00042     MOVZBL  6(R2), R0                   ;  1499
            02               A2        50   A2 00046     SUBW2   R0, 2(R2)                   ;
                          0000' CF   D7   00049 3$:      DECL    PORT_COUNT                  ;  1501
                             02   A2   B5   0004D         TSTW    2(R2)                       ;  1502
                                0D   12   00050         BNEQ    4$                          ;
```

```
                          0000'  CF   D6 00059          INCL    CNT_RP_ZERO                                 ; 1507
                            04   AC   DD 0005D          PUSHL   RTE                                         ; 1508
                0000V       CF        01   FB 00060     CALLS   #1, DEALLOCATE_BUF                          ;
                                      04 00065          RET                                                 ; 1503
                      50        02    A2   3C 00066 4$:  MOVZWL  2(R2), R0                                  ; 1513
                0000'       CF        50   C0 0006A     ADDL2   R0, CNT_RP_CHAR                             ;
                0000'       DF        04   BC   0E 0006F  INSQUE  @RTE, @WRITE_COMMAND_Q+4                   ; 1515
                0000V       CF        00   FB 00075     CALLS   #0, WRITE_COMMAND                           ; 1517
                                      04 0007A          RET                                                 ; 1519

; Routine Size:  123 bytes,    Routine Base:  $CODE$ + 0350
```

```
;   805          1520  1 ROUTINE write_command : NOVALUE =
;   806          1521  1
;   807          1522  1 !++
;   808          1523  1 !
;   809          1524  1 ! FUNCTIONAL DESCRIPTION:
;   810          1525  1 !
;   811          1526  1 !
;   812          1527  1 !
;   813          1528  1 ! CALLING SEQUENCE:
;   814          1529  1 !
;   815          1530  1 ! INPUT PARAMETERS:
;   816          1531  1 !       NONE
;   817          1532  1 !
;   818          1533  1 ! IMPLICIT INPUTS:
;   819          1534  1 !       NONE
;   820          1535  1 !
;   821          1536  1 ! OUTPUT PARAMETERS:
;   822          1537  1 !       NONE
;   823          1538  1 !
;   824          1539  1 ! IMPLICIT OUTPUTS:
;   825          1540  1 !       NONE
;   826          1541  1 !
;   827          1542  1 ! ROUTINE VALUE:
;   828          1543  1 !       NONE
;   829          1544  1 !
;   830          1545  1 ! SIDE EFFECTS:
;   831          1546  1 !       NONE
;   832          1547  1 !
;   833          1548  1 !--
;   834          1549  1
;   835          1550  2 BEGIN
;   836          1551  2
;   837          1552  2 LOCAL
;   838          1553  2     rte:        VOLATILE REF $BBLOCK;
;   839          1554  2
;   840          1555  2 !
;   841          1556  2 ! Check for multiple writes active.
;   842          1557  2 !
;   843          1558  2 IF .write_active_count GEQ 3 THEN RETURN;
;   844          1559  2
;   845          1560  2 !
;   846          1561  2 ! Check for empty queue
;   847          1562  2 !
;   848          1563  2 IF .write_command_q [0] EQL write_command_q [0] THEN
;   849          1564  3     BEGIN
;   850          1565  3     RETURN;
;   851          1566  2     END;
;   852          1567  2
;   853          1568  2 write_active_count = .write_active_count + 1;
;   854          1569  2 REMQUE (.write_command_q [0], rte);
;   855          1570  2
;   856          1571  3 BEGIN
;   857          1572  3 BIND
;   858          1573  3     IOSB_WORD = rte [rte$q_iosb] : VECTOR [,WORD],
;   859          1574  3     IOSB_BYTE = rte [rte$q_iosb] : VECTOR [,BYTE];
;   860          1575  3
;   861        P 1576  3     quit_if_error (
```

```
;  862          P 1577  3          $QIO (CHAN      = .command_chan,
;  863          P 1578  3                FUNC      = io$_writevblk,
;  864          P 1579  3                ASTADR    = Write_command_ast,
;  865          P 1580  3                ASTPRM    = .rte,
;  866          P 1581  3                IOSB      = rte [rte$q_iosb],
;  867          P 1582  3                P1        = rte [rte$t_buf],
;  868          P 1583  3                P2        = .iosb_word [w_riosb_datasize],
;  869          P 1584  3                P4        = 0)                    ! 0 forces no carriage control
;  870            1585  3                );
;  871            1586  2 END; ! BIND
;  872            1587  2
;  873            1588  1 END;                            ! End of routine write_command


                                        000C 00000 WRITE_COMMAND:
                                                   .WORD   Save R2,R3                                  ; 1520
                                53      0000'  CF  9E 00002         MOVAB   WRITE_COMMAND_Q, R3         ;
                                5E             04  C2 00007         SUBL2   #4, SP                      ;
                                03        E8   A3  D1 0000A         CMPL    WRITE_ACTIVE_COUNT, #3      ; 1558
                                               58  18 0000E         BGEQ    1$                          ;
                                50             63  9E 00010         MOVAB   WRITE_COMMAND_Q, R0         ; 1563
                                50             63  D1 00013         CMPL    WRITE_COMMAND_Q, R0         ;
                                               50  13 00016         BEQL    1$                          ;
                                           E8  A3  D6 00018         INCL    WRITE_ACTIVE_COUNT          ; 1568
                                6E         00  B3  0F 0001B         REMQUE  @WRITE_COMMAND_Q, RTE       ; 1569
                          50                6E      0C  C1 0001F    ADDL3   #12, RTE, R0                ; 1573
                                                   7E  7C 00023     CLRQ    -(SP)                       ; 1585
                                                   7E  7C 00025     CLRQ    -(SP)                       ;
                                        7E         02  A0 00027     MOVZWL  2(R0), -(SP)                ;
                                  7E          14   AE      14  C1 0002B    ADDL3   #20, RTE, -(SP)         ;
                                                       18  AE DD 00030    PUSHL   RTE                     ;
                                                  0000V    CF 9F 00033    PUSHAB  WRITE_COMMAND_AST       ;
                                  7E          20   AE      0C  C1 00037    ADDL3   #12, RTE, -(SP)         ;
                                                       30  DD 0003C    PUSHL   #48                     ;
                                                   7E      DC  A3 3C 0003E    MOVZWL  COMMAND_CHAN, -(SP)     ;
                                                               7E D4 00042    CLRL    -(SP)               ;
                                     00000000G     00            0C FB 00044    CALLS   #12, SYS$QIO        ;
                                                   52            50 D0 0004B    MOVL    R0, STATUS          ;
                                                   17            52 E8 0004E    BLBS    STATUS, 1$          ;
                                     00000000G     00            01 D0 00051    MOVL    #1, WAKEFLAG        ;
                                                                 7E 7C 00058    CLRQ    -(SP)               ;
                                     00000000G     00            02 FB 0005A    CALLS   #2, SYS$WAKE        ;
                                     00000000G     00            52 D0 00061    MOVL    STATUS, RETSTATUS   ;
                                                                 04 00068 1$:   RET                         ; 1588

; Routine Size:  105 bytes,    Routine Base:  $CODE$ + 03CB

;  874           1589  1
;  875           1590  1
```

```
;  877     1591  1 ROUTINE Write_command_ast (rte: REF $BBLOCK) : NOVALUE =
;  878     1592  1
;  879     1593  1 !++
;  880     1594  1 !
;  881     1595  1 ! FUNCTIONAL DESCRIPTION:
;  882     1596  1 !
;  883     1597  1 !       Write to command terminal has completed, delete rte buffer.
;  884     1598  1 !
;  885     1599  1 ! CALLING SEQUENCE:
;  886     1600  1 !
;  887     1601  1 ! INPUT PARAMETERS:
;  888     1602  1 !       NONE
;  889     1603  1 !
;  890     1604  1 ! IMPLICIT INPUTS:
;  891     1605  1 !       NONE
;  892     1606  1 !
;  893     1607  1 ! OUTPUT PARAMETERS:
;  894     1608  1 !       NONE
;  895     1609  1 !
;  896     1610  1 ! IMPLICIT OUTPUTS:
;  897     1611  1 !       NONE
;  898     1612  1 !
;  899     1613  1 ! ROUTINE VALUE:
;  900     1614  1 !       NONE
;  901     1615  1 !
;  902     1616  1 ! SIDE EFFECTS:
;  903     1617  1 !       NONE
;  904     1618  1 !
;  905     1619  1 !--
;  906     1620  1
;  907     1621  2 BEGIN
;  908     1622  2 BIND
;  909     1623  2     CHAR_ARRAY = rte [rte$t_buf] : VECTOR [,byte],
;  910     1624  2     IOSB_WORD = rte [rte$q_iosb] : VECTOR [,word],
;  911     1625  2     IOSB_BYTE = rte [rte$q_iosb] : VECTOR [,byte];
;  912     1626  2 LOCAL
;  913     1627  2     qio_size;
;  914     1628  2 !
;  915     1629  2 ! write to screen is complete, decrement count and call routine to
;  916     1630  2 ! check for more pending writes.
;  917     1631  2 !
;  918     1632  2 write_active_count = .write_active_count - 1;
;  919     1633  2 write_command ();
;  920     1634  2
;  921     1635  2 IF (.cterm_flag AND flg$m_logging) NEQ 0 THEN   ! if logging
;  922     1636  3     BEGIN
;  923     1637  3     qio_size = .iosb_word [1];  !***
;  924     1638  3     INCR i FROM 0 TO .qio_Size - 1 DO
;  925     1639  4         BEGIN
;  926     1640  4         IF NOT (.char_array [.i] EQL char$c_lf) THEN
;  927     1641  4             IF (.char_array [.i] EQL char$c_cr) OR (.log_count GEQ 512) THEN
;  928     1642  5                 BEGIN
;  929     1643  5                 IF (.log_count GTR 0) THEN
;  930     1644  5                     rtlog$write_string (log_buffer, .log_count);
;  931     1645  5                 log_count = 0;
;  932     1646  5                 END
;  933     1647  4             ELSE
```

```
;  934          1648  5                  BEGIN
;  935          1649  5                  log_buffer [.log_count] = .char_array [.i];
;  936          1650  5                  log_count = .log_count + 1;
;  937          1651  4                  END;
;  938          1652  3              END;
;  939          1653  2          END;
;  940          1654  2
;  941          1655  2  deallocate_buf (.rte);
;  942          1656  2  !***port_count = .port_count - 1;
;  943          1657  2
;  944          1658  2
;  945          1659  2  !IF .read_pending THEN
;  946          1660  2  cnt_wr_ast = .cnt_wr_ast + 1;
;  947          1661  2  !read_port_timed (0);
;  948          1662  2
;  949          1663  1  END;                                    ! End of routine Write_command_ast


                                        003C 00000 WRITE_COMMAND_AST:
                                                            .WORD   Save R2,R3,R4,R5                        ; 1591
                     55      0000'  CF  9E 00002            MOVAB   LOG_COUNT, R5                           ;
           52        04      AC         14  C1 00007        ADDL3   #20, RTE, R2                            ; 1623
           53        04      AC         0C  C1 0000C        ADDL3   #12, RTE, R3                            ; 1624
                                   10    A5  D7 00011        DECL    WRITE_ACTIVE_COUNT                      ; 1632
                  FF7E      CF         00  FB 00014        CALLS   #0, WRITE_COMMAND                       ; 1633
           43 00000000G     00         03  E1 00019        BBC     #3, CTERM_FLAG, 6$                      ; 1635
                     54        02      A3  3C 00021        MOVZWL  2(R3), QIO_SIZE                         ; 1637
                     53                01  CE 00025        MNEGL   #1, I                                   ; 1638
                                       36  11 00028        BRB     5$                                      ;
                     0A              6342  91 0002A 1$:     CMPB    (I)[R2], #10                            ; 1640
                                       30  13 0002E        BEQL    5$                                      ;
                     50                65  D0 00030        MOVL    LOG_COUNT, R0                           ; 1643
                     0D              6342  91 00033        CMPB    (I)[R2], #13                            ; 1641
                                       09  13 00037        BEQL    2$                                      ;
              00000200      8F         65  D1 00039        CMPL    LOG_COUNT, #512                         ;
                                       15  19 00040        BLSS    4$                                      ;
                                       50  D5 00042 2$:     TSTL    R0                                      ; 1643
                                       0D  15 00044        BLEQ    3$                                      ;
                                       50  DD 00046        PUSHL   R0                                      ; 1644
                                 FE00  C5  9F 00048        PUSHAB  LOG_BUFFER                              ;
              00000000G      00         02  FB 0004C        CALLS   #2, RTLOG$WRITE_STRING                  ;
                                       65  D4 00053 3$:     CLRL    LOG_COUNT                               ; 1645
                                       09  11 00055        BRB     5$                                      ; 1641
                  FE00 C540          6342  90 00057 4$:     MOVB    (I)[R2], LOG_BUFFER[R0]                 ; 1649
                                       65  D6 0005E        INCL    LOG_COUNT                               ; 1650
                        C6      53         54  F2 00060 5$:     AOBLSS  QIO_SIZE, I, 1$                 ; 1638
                                   04    AC  DD 00064 6$:     PUSHL   RTE                                     ; 1655
                  0000V         CF         01  FB 00067        CALLS   #1, DEALLOCATE_BUF                      ;
                                 FDFC  C5  D6 0006C        INCL    CNT_WR_AST                              ; 1660
                                           04 00070        RET                                             ; 1663

; Routine Size:  113 bytes,    Routine Base:  $CODE$ + 0434
```

; 950 1664 1

```
;  952          1665  1 ROUTINE Read_sysinput =
;  953          1666  1
;  954          1667  1 !++
;  955          1668  1 !
;  956          1669  1 ! FUNCTIONAL DESCRIPTION:
;  957          1670  1 !
;  958          1671  1 !       Read from SYS$INPUT so as to fake reading from SYS$COMMAND.
;  959          1672  1 !       This allows SET HOST from a command file to read commands
;  960          1673  1 !       directly from that file.
;  961          1674  1 !
;  962          1675  1 ! CALLING SEQUENCE:
;  963          1676  1 !
;  964          1677  1 !       Read_sysinput
;  965          1678  1 !
;  966          1679  1 ! INPUT PARAMETERS:
;  967          1680  1 !       NONE
;  968          1681  1 !
;  969          1682  1 ! IMPLICIT INPUTS:
;  970          1683  1 !
;  971          1684  1 !       File opened by RTPAD module.
;  972          1685  1 !
;  973          1686  1 ! OUTPUT PARAMETERS:
;  974          1687  1 !       NONE
;  975          1688  1 !
;  976          1689  1 ! IMPLICIT OUTPUTS:
;  977          1690  1 !       NONE
;  978          1691  1 !
;  979          1692  1 ! ROUTINE VALUE:
;  980          1693  1 !       NONE
;  981          1694  1 !
;  982          1695  1 ! SIDE EFFECTS:
;  983          1696  1 !       NONE
;  984          1697  1 !
;  985          1698  1 !--
;  986          1699  1
;  987          1700  2 BEGIN
;  988          1701  2
;  989          1702  2 RETURN (ss$_normal);
;  990          1703  2
;  991          1704  1 END;                                    ! End of routine



                                        0000 00000 READ_SYSINPUT:
                                                   .WORD   Save nothing                 ; 1665
                              50        01  D0 00002         MOVL    #1, R0             ; 1702
                                        04 00005             RET                        ; 1704

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 04A5

;  992          1705  1
```

```
;  994          1706  1 ROUTINE read_port_mbx (rte: REF $BBLOCK) : NOVALUE =
;  995          1707  1
;  996          1708  1 !++
;  997          1709  1 !
;  998          1710  1 ! FUNCTIONAL DESCRIPTION:
;  999          1711  1 !
; 1000          1712  1 ! Read from mailbox assocaiated with port channel.  This mailbox will provide
; 1001          1713  1 ! unsolicited data notification
; 1002          1714  1 !
; 1003          1715  1 ! CALLING SEQUENCE:
; 1004          1716  1 !
; 1005          1717  1 ! INPUT PARAMETERS:
; 1006          1718  1 !       NONE
; 1007          1719  1 !
; 1008          1720  1 ! IMPLICIT INPUTS:
; 1009          1721  1 !       NONE
; 1010          1722  1 !
; 1011          1723  1 ! OUTPUT PARAMETERS:
; 1012          1724  1 !       NONE
; 1013          1725  1 !
; 1014          1726  1 ! IMPLICIT OUTPUTS:
; 1015          1727  1 !       NONE
; 1016          1728  1 !
; 1017          1729  1 ! ROUTINE VALUE:
; 1018          1730  1 !       NONE
; 1019          1731  1 !
; 1020          1732  1 ! SIDE EFFECTS:
; 1021          1733  1 !       NONE
; 1022          1734  1 !
; 1023          1735  1 !--
; 1024          1736  1
; 1025          1737  2 BEGIN
; 1026          1738  2
; 1027        P 1739  2 Quit_if_error (
; 1028        P 1740  2     $QIO (CHAN  = .port_mbx_chan,
; 1029        P 1741  2           FUNC  = io$_readvblk,
; 1030        P 1742  2           ASTADR  = Read_mbx_ast,
; 1031        P 1743  2           ASTPRM  = .rte,
; 1032        P 1744  2           IOSB  = rte [rte$q_iosb],
; 1033        P 1745  2           P1  = rte [rte$t_buf],
; 1034        P 1746  2           P2  = rte$c_buflen)
; 1035          1747  2           );
; 1036          1748  2
; 1037          1749  1 END;                                    ! End of routine read_port_mbx


                                        0004 00000 READ_PORT_MBX:
                                                           .WORD   Save R2                                 ; 1706
                                             7E  7C 00002          CLRQ    -(SP)                          ; 1747
                                             7E  7C 00004          CLRQ    -(SP)                          ;
                                    7E       50  8F  9A 00006      MOVZBL  #80, -(SP)                     ;
                          7E        04  AC       14  C1 0000A      ADDL3   #20, RTE, -(SP)                ;
                                          04  AC  DD 0000F        PUSHL   RTE                            ;
                                    0000V CF  9F 00012            PUSHAB  READ_MBX_AST                   ;
```

```
          7E        04   AC          0C  C1 00016         ADDL3   #12, RTE, -(SP)                    ;
                                     31  DD 0001B         PUSHL   #49                                ;
                         7E    0000' CF  3C 0001D         MOVZWL  PORT_MBX_CHAN, -(SP)               ;
                                     7E  D4 00022         CLRL    -(SP)                              ;
              00000000G  00          0C  FB 00024         CALLS   #12, SYS$QIO                       ;
                         52          50  D0 0002B         MOVL    R0, STATUS                         ;
                         17          52  E8 0002E         BLBS    STATUS, 1$                         ;
              00000000G  00          01  D0 00031         MOVL    #1, WAKEFLAG                       ;
                                     7E  7C 00038         CLRQ    -(SP)                              ;
              00000000G  00          02  FB 0003A         CALLS   #2, SYS$WAKE                       ;
              00000000G  00          52  D0 00041         MOVL    STATUS, RETSTATUS                  ;
                                         04 00048 1$:     RET                                        ; 1749

; Routine Size:  73 bytes,    Routine Base:  $CODE$ + 04AB


; 1038          1750  1
```

```
;  1040          1751  1 ROUTINE read_mbx_ast (rte: REF $BBLOCK) : NOVALUE =
;  1041          1752  1
;  1042          1753  1 !++
;  1043          1754  1 !
;  1044          1755  1 ! FUNCTIONAL DESCRIPTION:
;  1045          1756  1 !
;  1046          1757  1 ! Receives AST from port mailbox.
;  1047          1758  1 !
;  1048          1759  1 ! CALLING SEQUENCE:
;  1049          1760  1 !
;  1050          1761  1 !       called as an AST routine.
;  1051          1762  1 !
;  1052          1763  1 ! INPUT PARAMETERS:
;  1053          1764  1 !
;  1054          1765  1 !       RTE - address of I/O block
;  1055          1766  1 !
;  1056          1767  1 ! IMPLICIT INPUTS:
;  1057          1768  1 !       NONE
;  1058          1769  1 !
;  1059          1770  1 ! OUTPUT PARAMETERS:
;  1060          1771  1 !       NONE
;  1061          1772  1 !
;  1062          1773  1 ! IMPLICIT OUTPUTS:
;  1063          1774  1 !       NONE
;  1064          1775  1 !
;  1065          1776  1 ! ROUTINE VALUE:
;  1066          1777  1 !       NONE
;  1067          1778  1 !
;  1068          1779  1 ! SIDE EFFECTS:
;  1069          1780  1 !       NONE
;  1070          1781  1 !
;  1071          1782  1 !--
;  1072          1783  1
;  1073          1784  2 BEGIN
;  1074          1785  2
;  1075          1786  2 BIND
;  1076          1787  2     msgcode = rte [rte$t_buf] : WORD;
;  1077          1788  2
;  1078          1789  2 read_pending = true;                          ! assume read won't happen
;  1079          1790  2
;  1080          1791  2 IF (.msgcode EQL msg$_trmunsolic) THEN
;  1081          1792  3     BEGIN
;  1082          1793  3     cnt_rp_uns = .cnt_rp_uns + 1;
;  1083          1794  3     read_port_timed (T);
;  1084          1795  2     END;
;  1085          1796  2
;  1086          1797  2 read_port_mbx (.rte);
;  1087          1798  2
;  1088          1799  1 END;                                  ! End of routine read_mbx_ast


                                0000 00000 READ_MBX_AST:
                                           .WORD   Save nothing                                    ; 1751
                50       04   AC          14  C1 00002        ADDL3   #20, RTE, R0               ; 1787
```

```
                          0000'  CF          01  DO 00007          MOVL    #1, READ_PENDING                        ; 1789
                                 01          60  B1 0000C          CMPW    (R0), #1                                ; 1791
                                             0B  12 0000F          BNEQ    1$                                      ;
                                       0000' CF  D6 00011          INCL    CNT_RP_UNS                              ; 1793
                                             01  DD 00015          PUSHL   #1                                      ; 1794
                          FDB3   CF          01  FB 00017          CALLS   #1, READ_PORT_TIMED                     ;
                                       04    AC  DD 0001C 1$:      PUSHL   RTE                                     ; 1797
                            94   AF          01  FB 0001F          CALLS   #1, READ_PORT_MBX                       ;
                                             04  00023          RET                                             ; 1799

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 04F4

; 1089          1800  1
; 1090          1801  1
```

```
;  1092          1802  1 ROUTINE allocate_buf (size) =
;  1093          1803  1
;  1094          1804  1 !++
;  1095          1805  1 !
;  1096          1806  1 ! FUNCTIONAL DESCRIPTION:
;  1097          1807  1 !
;  1098          1808  1 !
;  1099          1809  1 !
;  1100          1810  1 ! CALLING SEQUENCE:
;  1101          1811  1 !
;  1102          1812  1 ! INPUT PARAMETERS:
;  1103          1813  1 !       NONE
;  1104          1814  1 !
;  1105          1815  1 ! IMPLICIT INPUTS:
;  1106          1816  1 !       NONE
;  1107          1817  1 !
;  1108          1818  1 ! OUTPUT PARAMETERS:
;  1109          1819  1 !       NONE
;  1110          1820  1 !
;  1111          1821  1 ! IMPLICIT OUTPUTS:
;  1112          1822  1 !       NONE
;  1113          1823  1 !
;  1114          1824  1 ! ROUTINE VALUE:
;  1115          1825  1 !       NONE
;  1116          1826  1 !
;  1117          1827  1 ! SIDE EFFECTS:
;  1118          1828  1 !       NONE
;  1119          1829  1 !
;  1120          1830  1 !--
;  1121          1831  1
;  1122          1832  2 BEGIN
;  1123          1833  2
;  1124          1834  2 LOCAL
;  1125          1835  2     buf : REF $BBLOCK,
;  1126          1836  2     status;
;  1127          1837  2
;  1128          1838  2 quit_if_error (LIB$GET_VM (size, buf));
;  1129          1839  2
;  1130          1840  2 buf [rte$w_size] = .size;
;  1131          1841  2
;  1132          1842  2 RETURN (.buf);
;  1133          1843  2
;  1134          1844  1 END;                                    ! End of routine


                                0004 00000 ALLOCATE_BUF:
                                           .WORD   Save R2                                 ; 1802
                          5E               04  C2 00002         SUBL2   #4, SP
                                           5E  DD 00005         PUSHL   SP                 ; 1838
                                     04    AC  9F 00007         PUSHAB  SIZE
                00000000G   00             02  FB 0000A         CALLS   #2, LIB$GET_VM
                            52             50  D0 00011         MOVL    R0, STATUS
                            1B             52  E8 00014         BLBS    STATUS, 1$
                00000000G   00             01  D0 00017         MOVL    #1, WAKEFLAG           ;
```

```
                                        7E  7C 0001E           CLRQ    -(SP)                                          ;
                        000C 00OG   00  02  FB 00020           CALLS   #2, SYS$WAKE                                   ;
                        00000000G   00  52  D0 00027           MOVL    STATUS, RETSTATUS                              ;
                                    50  52  D0 0002E           MOVL    STATUS, R0                                     ;
                                        04 00031               RET                                                    ;
                                    50  6E  D0 00032 1$:       MOVL    BUF, R0                                        ; 1840
                                08  A0      04  AC  B0 00035   MOVW    SIZE, 8(R0)                                    ;
                                        04 0003A               RET                                                    ; 1844

; Routine Size:  59 bytes,    Routine Base:  $CODE$ + 0518

; 1135          1845  1
```

```
;  1137          1846  1 ROUTINE deallocate_buf (buf: REF $BBLOCK) : NOVALUE =
;  1138          1847  1
;  1139          1848  1 !++
;  1140          1849  1 !
;  1141          1850  1 ! FUNCTIONAL DESCRIPTION:
;  1142          1851  1 !
;  1143          1852  1 !
;  1144          1853  1 !
;  1145          1854  1 ! CALLING SEQUENCE:
;  1146          1855  1 !
;  1147          1856  1 ! INPUT PARAMETERS:
;  1148          1857  1 !       NONE
;  1149          1858  1 !
;  1150          1859  1 ! IMPLICIT INPUTS:
;  1151          1860  1 !       NONE
;  1152          1861  1 !
;  1153          1862  1 ! OUTPUT PARAMETERS:
;  1154          1863  1 !       NONE
;  1155          1864  1 !
;  1156          1865  1 ! IMPLICIT OUTPUTS:
;  1157          1866  1 !       NONE
;  1158          1867  1 !
;  1159          1868  1 ! ROUTINE VALUE:
;  1160          1869  1 !       NONE
;  1161          1870  1 !
;  1162          1871  1 ! SIDE EFFECTS:
;  1163          1872  1 !       NONE
;  1164          1873  1 !
;  1165          1874  1 !--
;  1166          1875  1
;  1167          1876  2 BEGIN
;  1168          1877  2
;  1169          1878  2 Quit_if_error (LIB$FREE_VM (buf [rte$w_size], buf));
;  1170          1879  2
;  1171          1880  1 END;                                     ! End of routine



                                        0004 00000 DEALLOCATE_BUF:
                                                   .WORD   Save R2                                  ; 1846
                                    04  AC  9F 00002           PUSHAB  BUF                         ; 1878
                                7E      04  AC     08  C1 00005           ADDL3   #8, BUF, -(SP)      ;
                             00000000G  00         02  FB 0000A           CALLS   #2, LIB$FREE_VM     ;
                                        52         50  D0 00011           MOVL    R0, STATUS          ;
                                        17         52  E8 00014           BLBS    STATUS, 1$          ;
                             00000000G  00         01  D0 00017           MOVL    #1, WAKEFLAG        ;
                                                   7E  7C 0001E           CLRQ    -(SP)               ;
                             00000000G  00         02  FB 00020           CALLS   #2, SYS$WAKE        ;
                             00000000G  00         52  D0 00027           MOVL    STATUS, RETSTATUS   ;
                                                   04  0002E 1$:         RET                         ; 1880

; Routine Size:  47 bytes,    Routine Base:  $CODE$ + 0553
```

```
;  1173          1881  1 ROUTINE sense_mode (chan, block) =
;  1174          1882  1
;  1175          1883  1 !++
;  1176          1884  1 !
;  1177          1885  1 ! FUNCTIONAL DESCRIPTION:
;  1178          1886  1 !
;  1179          1887  1 !
;  1180          1888  1 !
;  1181          1889  1 ! CALLING SEQUENCE:
;  1182          1890  1 !
;  1183          1891  1 ! INPUT PARAMETERS:
;  1184          1892  1 !       NONE
;  1185          1893  1 !
;  1186          1894  1 ! IMPLICIT INPUTS:
;  1187          1895  1 !       NONE
;  1188          1896  1 !
;  1189          1897  1 ! OUTPUT PARAMETERS:
;  1190          1898  1 !       NONE
;  1191          1899  1 !
;  1192          1900  1 ! IMPLICIT OUTPUTS:
;  1193          1901  1 !       NONE
;  1194          1902  1 !
;  1195          1903  1 ! ROUTINE VALUE:
;  1196          1904  1 !       NONE
;  1197          1905  1 !
;  1198          1906  1 ! SIDE EFFECTS:
;  1199          1907  1 !       NONE
;  1200          1908  1 !
;  1201          1909  1 !--
;  1202          1910  1
;  1203          1911  2 BEGIN
;  1204          1912  2 LOCAL status;
;  1205          1913  2
;  1206          1914  2 status =
;  1207        P 1915  2     $QIOW (CHAN = .chan,
;  1208        P 1916  2          FUNC     = io$_sensemode,
;  1209        P 1917  2          P1       = .block,
;  1210          1918  2          P2       = 12);
;  1211          1919  2
;  1212          1920  2 RETURN (.status);
;  1213          1921  2
;  1214          1922  1 END;                                    ! End of routine sense_mode


                                                .EXTRN  SYS$QIOW

                                0000 00000 SENSE_MODE:
                                                .WORD   Save nothing                    ; 1881
                                     7E  7C 00002       CLRQ    -(SP)                   ; 1918
                                     7E  7C 00004       CLRQ    -(SP)                   ;
                                     0C  DD 00006       PUSHL   #12                     ;
                                08   AC  DD 00008       PUSHL   BLOCK                   ;
                                     7E  7C 0000B       CLRQ    -(SP)                   ;
                           7E        27  7D 0000D       MOVQ    #39, -(SP)              ;
                                04   AC  DD 00010       PUSHL   CHAN                    ;
                                     7E  D4 00013       CLRL    -(SP)                   ;
```

```
                          00000000G  00          0C  FB 00015          CALLS   #12, SYS$QIOW                        ;
                                                    04 0001C          RET                                         ; 1922

; Routine Size:  29 bytes,    Routine Base:  $CODE$ + 0582

; 1215          1923  1
; 1216          1924  1
```

```
;  1218          1925  1 ROUTINE SET_PASSTHRU_MODE (chan, char: REF VECTOR, nobrdcst) =
;  1219          1926  1
;  1220          1927  1 !++
;  1221          1928  1 !
;  1222          1929  1 ! FUNCTIONAL DESCRIPTION:
;  1223          1930  1 !
;  1224          1931  1 !       Sets PASSTHRU, TTSYNC, HOSTSYNC, and (optionally) NOBRDCST.
;  1225          1932  1 !
;  1226          1933  1 ! CALLING SEQUENCE:
;  1227          1934  1 !
;  1228          1935  1 ! INPUT PARAMETERS:
;  1229          1936  1 !       NONE
;  1230          1937  1 !
;  1231          1938  1 ! IMPLICIT INPUTS:
;  1232          1939  1 !       NONE
;  1233          1940  1 !
;  1234          1941  1 ! OUTPUT PARAMETERS:
;  1235          1942  1 !       NONE
;  1236          1943  1 !
;  1237          1944  1 ! IMPLICIT OUTPUTS:
;  1238          1945  1 !       NONE
;  1239          1946  1 !
;  1240          1947  1 ! ROUTINE VALUE:
;  1241          1948  1 !       NONE
;  1242          1949  1 !
;  1243          1950  1 ! SIDE EFFECTS:
;  1244          1951  1 !       NONE
;  1245          1952  1 !
;  1246          1953  1 !--
;  1247          1954  1
;  1248          1955  2 BEGIN
;  1249          1956  2
;  1250          1957  2 LOCAL
;  1251          1958  2     status,
;  1252          1959  2     new_char:   VECTOR [3];
;  1253          1960  2
;  1254          1961  2 new_char [0] = .char [0];
;  1255          1962  2 new_char [1] = .char [1] OR TT$M_TTSYNC OR TT$M_HOSTSYNC;
;  1256          1963  2 new_char [2] = .char [2] OR TT2$M_PASTHRU;
;  1257          1964  2
;  1258          1965  2 If .nobrdcst THEN
;  1259          1966  2     new_char [1] = .new_char [1] OR TT$M_NOBRDCST;
;  1260          1967  2
;  1261        P 1968  2 status = $QIOW ( CHAN = .chan,
;  1262        P 1969  2          FUNC    = io$_setmode,
;  1263        P 1970  2          P1      = new_char,
;  1264          1971  2          P2      = 12);
;  1265          1972  2
;  1266          1973  2 RETURN (.status);
;  1267          1974  2
;  1268          1975  1 END;                                    ! End of routine SET_PASSTHRU_MODE


                                                0000 00000 SET_PASSTHRU_MODE:
```

```
                                                             .WORD   Save nothing                          ; 1925
                         5E              08  C2 00002        SUBL2   #8, SP                                ;
                         50         08   AC  D0 00005        MOVL    CHAR, R0                              ; 1961
                                         60  DD 00009        PUSHL   (R0)                                  ;
      04    AE      04   A0              30  C9 0000B        BISL3   #48, 4(R0), NEW_CHAR+4                ; 1962
      08    AE      08   A0 00040000     8F  C9 00011        BISL3   #262144, 8(R0), NEW_CHAR+8            ; 1963
                         04         0C   AC  E9 0001B        BLBC    NOBRDCST, 1$                          ; 1965
                    06   AE              02  88 0001F        BISB2   #2, NEW_CHAR+4                        ; 1966
                                         7E  7C 00023 1$:    CLRQ    -(SP)                                 ; 1971
                                         7E  7C 00025        CLRQ    -(SP)                                 ;
                                         0C  DD 00027        PUSHL   #12                                   ;
                                    14   AE  9F 00029        PUSHAB  NEW_CHAR                              ;
                                         7E  7C 0002C        CLRQ    -(SP)                                 ;
                         7E              23  7D 0002E        MOVQ    #35, -(SP)                            ;
                                    04   AC  DD 00031        PUSHL   CHAN                                  ;
                                         7E  D4 00034        CLRL    -(SP)                                 ;
               00000000G 00              0C  FB 00036        CALLS   #12, SYS$QIOW                         ;
                                             04 0003D        RET                                           ; 1975

; Routine Size:  62 bytes,    Routine Base:  $CODE$ + 059F
```

```
;  1270          1976  1 ROUTINE read_delay_ast : NOVALUE =
;  1271          1977  1
;  1272          1978  1 !++
;  1273          1979  1 !
;  1274          1980  1 ! FUNCTIONAL DESCRIPTION:
;  1275          1981  1 !
;  1276          1982  1 !       AST that fires and requests read from port.
;  1277          1983  1 !
;  1278          1984  1 ! CALLING SEQUENCE:
;  1279          1985  1 !
;  1280          1986  1 ! INPUT PARAMETERS:
;  1281          1987  1 !       NONE
;  1282          1988  1 !
;  1283          1989  1 ! IMPLICIT INPUTS:
;  1284          1990  1 !       NONE
;  1285          1991  1 !
;  1286          1992  1 ! OUTPUT PARAMETERS:
;  1287          1993  1 !       NONE
;  1288          1994  1 !
;  1289          1995  1 ! IMPLICIT OUTPUTS:
;  1290          1996  1 !       NONE
;  1291          1997  1 !
;  1292          1998  1 ! ROUTINE VALUE:
;  1293          1999  1 !       NONE
;  1294          2000  1 !
;  1295          2001  1 ! SIDE EFFECTS:
;  1296          2002  1 !       NONE
;  1297          2003  1 !
;  1298          2004  1 !--
;  1299          2005  1
;  1300          2006  2 BEGIN
;  1301          2007  2
;  1302          2008  2 timer_pending = false;
;  1303          2009  2 cnt_rp_timer = .cnt_rp_timer + 1;
;  1304          2010  2 read_port_timed (1);
;  1305          2011  2
;  1306          2012  1 END;                    ! of routine read_delay_ast


                                        0000 00000 READ_DELAY_AST:
                                                           .WORD   Save nothing                    ;  1976
                                0000'   CF   D4 00002      CLRL    TIMER_PENDING                   ;  2008
                                0000'   CF   D6 00006      INCL    CNT_RP_TIMER                    ;  2009
                                        01   DD 0000A      PUSHL   #1                              ;  2010
                            FCD5    CF      01   FB 0000C      CALLS   #1, READ_PORT_TIMED         ;
                                            04 00011       RET                                     ;  2012

; Routine Size:  18 bytes,    Routine Base:  $CODE$ + 05DD
```

```
;  1308          2013  1 ROUTINE exit_routine =
;  1309          2014  1
;  1310          2015  1 !++
;  1311          2016  1 !
;  1312          2017  1 ! FUNCTIONAL DESCRIPTION:
;  1313          2018  1 !
;  1314          2019  1 !
;  1315          2020  1 !
;  1316          2021  1 ! CALLING SEQUENCE:
;  1317          2022  1 !
;  1318          2023  1 ! INPUT PARAMETERS:
;  1319          2024  1 !       NONE
;  1320          2025  1 !
;  1321          2026  1 ! IMPLICIT INPUTS:
;  1322          2027  1 !       NONE
;  1323          2028  1 !
;  1324          2029  1 ! OUTPUT PARAMETERS:
;  1325          2030  1 !       NONE
;  1326          2031  1 !
;  1327          2032  1 ! IMPLICIT OUTPUTS:
;  1328          2033  1 !       NONE
;  1329          2034  1 !
;  1330          2035  1 ! ROUTINE VALUE:
;  1331          2036  1 !       NONE
;  1332          2037  1 !
;  1333          2038  1 ! SIDE EFFECTS:
;  1334          2039  1 !       NONE
;  1335          2040  1 !
;  1336          2041  1 !--
;  1337          2042  1
;  1338          2043  2 BEGIN
;  1339          2044  2
;  1340          2045  2 $CANCEL (chan = .command_chan);
;  1341          2046  2
;  1342        P 2047  2 $QIOW ( CHAN = .command_chan,
;  1343        P 2048  2         FUNC    = io$_setmode,
;  1344        P 2049  2         P1      = command_char,
;  1345          2050  2         P2      = 12);
;  1346          2051  2
;  1347          2052  2 dump_stats();
;  1348          2053  2
;  1349          2054  2 RETURN (.retstatus);
;  1350          2055  2
;  1351          2056  1 END;                                   ! End of routine Exit_routine


                                                .EXTRN  SYS$CANCEL

                                0000 00000 EXIT_ROUTINE:
                                                .WORD   Save nothing                    ; 2013
                          7E    0000'  CF  3C 00002           MOVZWL  COMMAND_CHAN, -(SP)     ; 2045
              00000000G   00           01  FB 00007           CALLS   #1, SYS$CANCEL          ;
                                       7E  7C 0000E           CLRQ    -(SP)                   ; 2050
                                       7E  7C 00010           CLRQ    -(SP)                   ;
                                       0C  DD 00012           PUSHL   #12                     ;
                                0000'  CF  9F 00014           PUSHAB  COMMAND_CHAR            ;
```

```
                                 7E  7C 00018          CLRQ    -(SP)                                                  ;
                          7E     23  7D 00C1A          MOVQ    #35, -(SP)                                             ;
                          7E 0000' CF  3C 0001D          MOVZWL  COMMAND_CHAN, -(SP)                                    ;
                                 7E  D4 00022          CLRL    -(SP)                                                  ;
                00000000G 00      0C  FB 00024          CALLS   #12, SYS$QIOW                                          ;
                    0000V CF      00  FB 0002B          CALLS   #0, DUMP_STATS                                         ; 2052
                          50 00000000G 00  D0 00030     MOVL    RETSTATUS, R0                                          ; 2054
                                       04 00037         RET                                                            ; 2056

; Routine Size:  56 bytes,    Routine Base:  $CODE$ + 05E7
```

```
;  1353          2057  1 ROUTINE dump_stats : NOVALUE =
;  1354          2058  2 BEGIN
;  1355          2059  2 LOCAL
;  1356          2060  2     outdesc:    VECTOR [2],
;  1357          2061  2     outbuf:     VECTOR [132,BYTE];
;  1358          2062  2
;  1359          2063  2 If (.rtlog_flags AND rtlog$m_banner) EQL 0
;  1360          2064  2 THEN    ! If logging
;  1361          2065  2     RETURN;
;  1362          2066  2
;  1363          2067  2 !
;  1364          2068  2 ! %% report stats
;  1365          2069  2 !
;  1366          2070  2 outdesc [1] = outbuf;
;  1367          2071  2     PRINT (%ASCID '%REM-I-STATS, runtime statistics:',0);
;  1368          2072  2     PRINT (%ASCID '!8<!UL!>  calls to read_port_timed',        .cnt_rp_timed);
;  1369          2073  2     PRINT (%ASCID '!8<!UL!>  characters output to screen'      .cnt_rp_char);
;  1370          2074  2     PRINT (%ASCID '!8<!UL!>  reads for zero characters',       .cnt_rp_zero);
;  1371          2075  2     PRINT (%ASCID '!8<!UL!>  calls to $SETIMR ',               .cnt_rp_timer);
;  1372          2076  2     PRINT (%ASCID '!8<!UL!>  unsol data hits',                 .cnt_rp_uns);
;  1373          2077  2     PRINT (%ASCID '!8<!UL!>  calls to RPT from rp_ast',        .cnt_rp_ast);
;  1374          2078  2     PRINT (%ASCID '!8<!UL!>  calls to write screen',           .cnt_wr_ast);
;  1375          2079  2     PRINT (%ASCID '!8<!UL!>  max outstanding port reads',      .max_port_coun');
;  1376          2080  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

72 20 2C 53 54 41 54 53 2D 49 2D 4D 45 52 25  00060 P.AAO:  .ASCII  \%REM-I-STATS, runtime statistics:\<0><0>
69 74 73 69 74 61 74 73 20 65 6D 69 74 6E 75  0006F
                        00 00 3A 73 63        0007E
                                 00           00083          .ASCII  <0>
                              010E0021        00084 P.AAN:   .LONG   17694753
                              00000000'       00088          .ADDRESS P.AAO
73 6C 6C 61 63 20 20 3E 21 4C 55 21 3C 38 21  0008C P.AAQ:  .ASCII  \!8<!UL!>  calls to read_port_timed\<0>
5F 74 72 6F 70 5F 64 61 65 72 20 6F 74 20     0009B
                        00 64 65 6D 69        000AA
                                 00           000AF          .ASCII  <0>
                              010E0022        000B0 P.AAP:   .LONG   17694754
                              00000000'       000B4          .ADDRESS P.AAQ
61 72 61 68 63 20 20 3E 21 4C 55 21 3C 38 21  000B8 P.AAS:  .ASCII  \!8<!UL!>  characters output to screen\<0>
6F 74 20 74 75 70 74 75 6F 20 73 72 65 74 63  000C7
                        00 6E 65 65 72 63 73  000D6
                                 00 00        000DE          .ASCII  <0><0>
                              010E0025        000E0 P.AAR:   .LONG   17694757
                              00000000'       000E4          .ADDRESS P.AAS
73 64 61 65 72 20 20 3E 21 4C 55 21 3C 38 21  000E8 P.AAU:  .ASCII  \!8<!UL!>  reads for zero characters\<0>
61 72 61 68 63 20 6F 72 65 7A 20 72 6F 66 20  000F7
                        00 73 72 65 74 63     00106
                              010E0023        0010C P.AAT:   .LONG   17694755
                              00000000'       00110          .ADDRESS P.AAU
73 6C 6C 61 63 20 20 3E 21 4C 55 21 3C 38 21  00114 P.AAW:  .ASCII  \!8<!UL!>  calls to $SETIMR \<0>
      00 20 52 4D 49 54 45 53 24 20 6F 74 20  00123
                              010E001B        00130 P.AAV:   .LONG   17694747
                              00000000'       00134          .ADDRESS P.AAW
6C 6F 73 6E 75 20 20 3E 21 4C 55 21 3C 38 21  00138 P.AAY:  .ASCII  \!8<!UL!>  unsol data hits\<0><0><0>
```

```
                00 00 00 73 74 69 68 20 61 74 61 64 20  00147
                                        010E0019        00154 P.AAX:    .LONG   17694745
                                        00000000'       00158           .ADDRESS P.AAY
73 6C 6C 61 63 20 20 3E 21 4C 55 21 3C 38 21            0015C P.ABA:    .ASCII  \.8<!UL!>  calls to RPT from rp_ast\<0>
70 72 20 6D 6F 72 66 20 54 50 52 20 6F 74 20            0016B
                                     00 74 73 61 5F     0017A
                                                 00     0017F           .ASCII  <0>
                                        010E0022        00180 P.AAZ:    .LONG   17694754
                                        00000000'       00184           .ADDRESS P.ABA
73 6C 6C 61 63 20 20 3E 21 4C 55 21 3C 38 21            00188 P.ABC:    .ASCII  \!8<!UL!>  calls to write screen\<0>
65 65 72 63 73 20 65 74 69 72 77 20 6F 74 20            00197
                                           00 6E        001A6
                                        010E001F        001A8 P.ABB:    .LONG   17694751
                                        00000000'       001AC           .ADDRESS P.ABC
6F 20 78 61 6D 20 20 3E 21 4C 55 21 3C 38 21            001B0 P.ABE:    .ASCII  \!8<!UL!>  max outstanding port reads\
74 72 6F 70 20 67 6E 69 64 6E 61 74 73 74 75            001BF
                                  73 64 61 65 72 20     001CE
                                        010E0024        001D4 P.ABD:    .LONG   17694756
                                        00000000'       001D8           .ADDRESS P.ABE


                                                                        .PSECT  $CODE$,NOWRT,2

                                                003C 00000 DUMP_STATS:
                                                                        .WORD   Save R2,R3,R4,R5                ; 2057
                                    55      0000'   CF  9E 00002        MOVAB   CNT_RP_TIMED, R5
                                    54 00000000G    00  9E 00007        MOVAB   LIB$PUT_OUTPUT, R4
                                    53 00000000G    00  9E 0000F        MOVAB   LIB$SIGNAL, R3
                                    52 00000000G    00  9E 00015        MOVAB   SYS$FAO, R2
                                    5E      FF74    CE  9E 0001C        MOVAB   -140(SP), SP
                                    01 00000000G    00  E8 00021        BLBS    RTLOG_FLAGS, 1$             ; 2063
                                                    04  00028        RET
                                FC  AD              6E  9E 00029 1$:    MOVAB   OUTBUF, OUTDESC+4           ; 2070
                                F8  AD          84  8F  9A 0002D        MOVZBL  #132, OUTDESC               ; 2071
                                                7E  D4 00032            CLRL    -(SP)
                                            F8  AD  9F 00034            PUSHAB  OUTDESC
                                            F8  AD  9F 00037            PUSHAB  OUTDESC
                                        0000'   CF  9F 0003A            PUSHAB  P.AAN
                                    62          04  FB 0003E            CALLS   #4, SYS$FAO
                                    05          50  E8 00041            BLBS    STATUS, 2$
                                                50  DD 00044            PUSHL   STATUS
                                    63          01  FB 00046            CALLS   #1, LIB$SIGNAL
                                            F8  AD  9F 00049 2$:         PUSHAB  OUTDESC
                                    64          01  FB 0004C            CALLS   #1, LIB$PUT_OUTPUT
                                F8  AD          84  8F  9A 0004F        MOVZBL  #132, OUTDESC               ; 2072
                                                65  DD 00054            PUSHL   CNT_RP_TIMED
                                            F8  AD  9F 00056            PUSHAB  OUTDESC
                                            F8  AD  9F 00059            PUSHAB  OUTDESC
                                        0000'   CF  9F 0005C            PUSHAB  P.AAP
                                    62          04  FB 00060            CALLS   #4, SYS$FAO
                                    05          50  E8 00063            BLBS    STATUS, 3$
                                                50  DD 00066            PUSHL   STATUS
                                    63          01  FB 00068            CALLS   #1, LIB$SIGNAL
                                            F8  AD  9F 0006B 3$:         PUSHAB  OUTDESC
                                    64          01  FB 0006E            CALLS   #1, LIB$PUT_OUTPUT
                                F8  AD          84  8F  9A 00071        MOVZBL  #132, OUTDESC               ; 2073
```

```
                04   A5   DD 00076          PUSHL   CNT_RP_CHAR
                F8   AD   9F 00079          PUSHAB  OUTDESC
                F8   AD   9F 0007C          PUSHAB  OUTDESC
              0000'  CF   9F 0007F          PUSHAB  P.AAR
          62                  04   FB 00083          CALLS   #4, SYS$FAO
          05                  50   E8 00086          BLBS    STATUS, 4$
                              50   DD 00089          PUSHL   STATUS
          63                  01   FB 0008B          CALLS   #1, LIB$SIGNAL
                F8   AD   9F 0008E 4$:      PUSHAB  OUTDESC
          64                  01   FB 00091          CALLS   #1, LIB$PUT_OUTPUT
    F8    AD        84   8F   9A 00094          MOVZBL  #132, OUTDESC                   2074
                    08   A5   DD 00099          PUSHL   CNT_RP_ZERO
                    F8   AD   9F 0009C          PUSHAB  OUTDESC
                    F8   AD   9F 0009F          PUSHAB  OUTDESC
                  0000'  CF   9F 000A2          PUSHAB  P.AAT
              62                  04   FB 000A6          CALLS   #4, SYS$FAO
              05                  50   E8 000A9          BLBS    STATUS, 5$
                                  50   DD 000AC          PUSHL   STATUS
              63                  01   FB 000AE          CALLS   #1, LIB$SIGNAL
                    F8   AD   9F 000B1 5$:      PUSHAB  OUTDESC
              64                  01   FB 000B4          CALLS   #1, LIB$PUT_OUTPUT
    F8    AD        84   8F   9A 000B7          MOVZBL  #132, OUTDESC                   2075
                    0C   A5   DD 000BC          PUSHL   CNT_RP_TIMER
                    F8   AD   9F 000BF          PUSHAB  OUTDESC
                    F8   AD   9F 000C2          PUSHAB  OUTDESC
                  0000'  CF   9F 000C5          PUSHAB  P.AAV
              62                  04   FB 000C9          CALLS   #4, SYS$FAO
              05                  50   E8 000CC          BLBS    STATUS, 6$
                                  50   DD 000CF          PUSHL   STATUS
              63                  01   FB 000D1          CALLS   #1, LIB$SIGNAL
                    F8   AD   9F 000D4 6$:      PUSHAB  OUTDESC
              64                  01   FB 000D7          CALLS   #1, LIB$PUT_OUTPUT
    F8    AD        84   8F   9A 000DA          MOVZBL  #132, OUTDESC                   2076
                    10   A5   DD 000DF          PUSHL   CNT_RP_UNS
                    F8   AD   9F 000E2          PUSHAB  OUTDESC
                    F8   AD   9F 000E5          PUSHAB  OUTDESC
                  0000'  CF   9F 000E8          PUSHAB  P.AAX
              62                  04   FB 000EC          CALLS   #4, SYS$FAO
              05                  50   E8 000EF          BLBS    STATUS, 7$
                                  50   DD 000F2          PUSHL   STATUS
              63                  01   FB 000F4          CALLS   #1, LIB$SIGNAL
                    F8   AD   9F 000F7 7$:      PUSHAB  OUTDESC
              64                  01   FB 000FA          CALLS   #1, LIB$PUT_OUTPUT
    F8    AD        84   8F   9A 000FD          MOVZBL  #132, OUTDESC                   2077
                    14   A5   DD 00102          PUSHL   CNT_RP_AST
                    F8   AD   9F 00105          PUSHAB  OUTDESC
                    F8   AD   9F 00108          PUSHAB  OUTDESC
                  0000'  CF   9F 0010B          PUSHAB  P.AAZ
              62                  04   FB 0010F          CALLS   #4, SYS$FAO
              05                  50   E8 00112          BLBS    STATUS, 8$
                                  50   DD 00115          PUSHL   STATUS
              63                  01   FB 00117          CALLS   #1, LIB$SIGNAL
                    F8   AD   9F 0011A 8$:      PUSHAB  OUTDESC
              64                  01   FB 0011D          CALLS   #1, LIB$PUT_OUTPUT
    F8    AD        84   8F   9A 00120          MOVZBL  #132, OUTDESC                   2078
                    18   A5   DD 00125          PUSHL   CNT_WR_AST
                    F8   AD   9F 00128          PUSHAB  OUTDESC
```

```
                        F8  AD  9F 00012B          PUSHAB  OUTDESC                                  ;
                      0000' CF  9F 00012E          PUSHAB  P.ABB                                    ;
                         62     04  FB 00132       CALLS   #4, SYS$FAO                              ;
                         05     50  E8 00135       BLBS    STATUS, 9$                               ;
                                50  DD 00138       PUSHL   STATUS                                   ;
                         63     01  FB 0013A       CALLS   #1, LIB$SIGNAL                           ;
                        F8  AD  9F 0013D 9$:    PUSHAB  OUTDESC                                  ;
                         64     01  FB 00140       CALLS   #1, LIB$PUT_OUTPUT                       ;
                    F8   AD     84  8F  9A 00143    MOVZBL  #132, OUTDESC                            ; 2079
                        FC  A5  DD 00148          PUSHL   MAX_PORT_COUNT                           ;
                        F8  AD  9F 0014B          PUSHAB  OUTDESC                                  ;
                        F8  AD  9F 0014E          PUSHAB  OUTDESC                                  ;
                      0000' CF  9F 00151          PUSHAB  P.ABD                                    ;
                         62     04  FB 00155       CALLS   #4, SYS$FAO                              ;
                         05     50  E8 00158       BLBS    STATUS, 10$                              ;
                                50  DD 0015B       PUSHL   STATUS                                   ;
                         63     01  FB 0015D       CALLS   #1, LIB$SIGNAL                           ;
                        F8  AD  9F 00160 10$:   PUSHAB  OUTDESC                                  ;
                         64     01  FB 00163       CALLS   #1, LIB$PUT_OUTPUT                       ;
                                    04  00166       RET                                              ; 2080

; Routine Size:  359 bytes,    Routine Base:  $CODE$ + 0627

; 1377          2081  1
; 1378          2082  1 END
; 1379          2083  0 ELUDOM


                                                                .EXTRN  LIB$SIGNAL

;                                PSECT SUMMARY
;
;      Name                      Bytes                         Attributes
;
;  $OWN$                           636  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          476  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         1934  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                            Library Statistics
;
;                                      -------- Symbols --------      Pages       Processing
;      File                            Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1     9776        33         0         581        00:00.7
```

```
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RTDTE/OBJ=OBJ$:RTDTE MSRC$:RTDTE/UPDATE=(ENH$:RTDTE)

; Size:         1934 code + 1112 data bytes
; Run Time:       00:27.3
; Elapsed Time:   01:08.2
; Lines/CPU Min.   4574
; Lexemes/CPU-Min: 37572
; Memory Used:  229 pages
; Compilation Complete
```

RTMSG
LIS

RSXRT
LIS

RTDEF
LIS

RTDTE
LIS

RTPAD
LIS

RTLOGIO
LIS