```
PPPPPPPPPPPP      HHH          HHH     000000000      NNN          NNN  EEEEEEEEEEEEEEE
PPPPPPPPPPPPP     HHH          HHH     000000000      NNN          NNN  EEEEEEEEEEEEEEE
PPPPPPPPPPPPP     HHH          HHH     000000000      NNN          NNN  EEEEEEEEEEEEEEE
PPP          PPP  HHH          HHH  000          000  NNN          NNN  EEE
PPP          PPP  HHH          HHH  000          000  NNN          NNN  EEE
PPP          PPP  HHH          HHH  000          000  NNN          NNN  EEE
PPP          PPP  HHH          HHH  000          000  NNNNNN       NNN  EEE
PPP          PPP  HHH          HHH  000          000  NNNNNN       NNN  EEE
PPP          PPP  HHH          HHH  000          000  NNNNNN       NNN  EEE
PPPPPPPPPPPP      HHHHHHHHHHHHHHHH  000          000  NNN   NNN    NNN  EEEEEEEEEEE
PPPPPPPPPPPP      HHHHHHHHHHHHHHHH  000          000  NNN   NNN    NNN  EEEEEEEEEEE
PPPPPPPPPPPP      HHHHHHHHHHHHHHHH  000          000  NNN   NNN    NNN  EEEEEEEEEEE
PPP               HHH          HHH  000          000  NNN      NNNNNN  EEE
PPP               HHH          HHH  000          000  NNN      NNNNNN  EEE
PPP               HHH          HHH  000          000  NNN      NNNNNN  EEE
PPP               HHH          HHH  000          000  NNN         NNN  EEE
PPP               HHH          HHH  000          000  NNN         NNN  EEE
PPP               HHH          HHH  000          000  NNN         NNN  EEE
PPP               HHH          HHH     000000000      NNN         NNN  EEEEEEEEEEEEEEE
PPP               HHH          HHH     000000000      NNN         NNN  EEEEEEEEEEEEEEE
PPP               HHH          HHH     000000000      NNN         NNN  EEEEEEEEEEEEEEE
```

IN
VO

```
IIIIII    NN       NN  PPPPPPPP    UU       UU  TTTTTTTTTT
IIIIII    NN       NN  PPPPPPPP    UU       UU  TTTTTTTTTT
  II      NN       NN  PP      PP  UU       UU      TT
  II      NN       NN  PP      PP  UU       UU      TT
  II      NNNN     NN  PP      PP  UU       UU      TT
  II      NNNN     NN  PP      PP  UU       UU      TT
  II      NN  NN   NN  PPPPPPPP    UU       UU      TT
  II      NN  NN   NN  PPPPPPPP    UU       UU      TT
  II      NN    NNNN  PP          UU       UU      TT
  II      NN    NNNN  PP          UU       UU      TT
  II      NN      NN  PP          UU       UU      TT          ....
  II      NN      NN  PP          UU       UU      TT          ....
IIIIII    NN      NN  PP          UUUUUUUUUU       TT          ....
IIIIII    NN      NN  PP          UUUUUUUUUU       TT          ....

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

```
;  1     0001  0 %title 'INPUT - Analyze and Act on Input'
;  2     0002  0         module input (
;  3     0003  1                         ident='V04-000') = begin
;  4     0004  1
;  5     0005  1 !
;  6     0006  1 !*****************************************************************************
;  7     0007  1 !*                                                                           *
;  8     0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;  9     0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
; 10     0010  1 !*  ALL RIGHTS RESERVED.                                                     *
; 11     0011  1 !*                                                                           *
; 12     0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 13     0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 14     0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
; 15     0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 16     0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 17     0017  1 !*  TRANSFERRED.                                                             *
; 18     0018  1 !*                                                                           *
; 19     0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 20     0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 21     0021  1 !*  CORPORATION.                                                             *
; 22     0022  1 !*                                                                           *
; 23     0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 24     0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
; 25     0025  1 !*                                                                           *
; 26     0026  1 !*                                                                           *
; 27     0027  1 !*****************************************************************************
; 28     0028  1 !
; 29     0029  1
; 30     0030  1 !++
; 31     0031  1 ! Facility:     VAX/VMS Telephone Facility, Analyze and Act on Input
; 32     0032  1 !
; 33     0033  1 ! Abstract:     This module receives all of the terminal input from
; 34     0034  1 !               the user and decides what to do with it.  It handles
; 35     0035  1 !               both conversation text and commands.
; 36     0036  1 !
; 37     0037  1 !
; 38     0038  1 ! Environment:
; 39     0039  1 !
; 40     0040  1 ! Author: Paul C. Anagnostopoulos, Creation Date: 10 November 1980
; 41     0041  1 !
; 42     0042  1 ! Modified By:
; 43     0043  1 !
; 44     0044  1 !       V03-001 PCA1020         Paul C. Anagnostopoulos 24-May-1983
; 45     0045  1 !               Support 8-bit characters.
; 46     0046  1 !               Add PHONE as a synonym for the DIAL command.
; 47     0047  1 !--
```

```
;    49          0048  1 %sbttl 'Module Declarations'
;    50          0049  1 !
;    51          0050  1 ! Libraries and Requires:
;    52          0051  1 !
;    53          0052  1
;    54          0053  1 library 'sys$library:starlet.l32';
;    55          0054  1 library 'sys$library:tpamac.l32';
;    56          0055  1 require 'phone-eq';
;    57          0384  1
;    58          0385  1 !
;    59          0386  1 ! Table of Contents:
;    60          0387  1 !
;    61          0388  1
;    62          0389  1 forward routine
;    63          0390  1         phn$kbd_route: novalue,
;    64          0391  1         phn$cmd_parse: novalue;
;    65          0392  1
;    66          0393  1 !
;    67          0394  1 ! External References:
;    68          0395  1 !
;    69          0396  1
;    70          0397  1 external routine
;    71          0398  1         lib$tparse: addressing_mode(general),
;    72          0399  1         phn$answer_cmd,
;    73          0400  1         phn$break_call,
;    74          0401  1         phn$conversing,
;    75          0402  1         phn$dial_cmd,
;    76          0403  1         phn$directory_cmd,
;    77          0404  1         phn$exit_cmd,
;    78          0405  1         phn$facsimile_cmd,
;    79          0406  1         phn$fresh_screen,
;    80          0407  1         phn$hangup_cmd,
;    81          0408  1         phn$help_cmd,
;    82          0409  1         phn$hold_cmd,
;    83          0410  1         phn$inform,
;    84          0411  1         phn$mail_cmd,
;    85          0412  1         phn$queue_smb,
;    86          0413  1         phn$reject_cmd,
;    87          0414  1         phn$show_text,
;    88          0415  1         phn$transcribe_cmd,
;    89          0416  1         phn$unhold_cmd;
;    90          0417  1
;    91          0418  1 external literal
;    92          0419  1         lib$_syntaxerr;
;    93          0420  1
;    94          0421  1 !
;    95          0422  1 ! Own Variables:
;    96          0423  1 !
;    97          0424  1
```

```
;    99          0425  1 %sbttl 'PHN$KBD_ROUTE - Initial Routing of Input'
;   100          0426  1 !++
;   101          0427  1 ! Functional Description:
;   102          0428  1 !       This steering message routine is invoked after we collect
;   103          0429  1 !       a bit of input from the keyboard.  It splits up the input
;   104          0430  1 !       into conversation text and command text and routes these
;   105          0431  1 !       to the appropriate steering routine for further processing.
;   106          0432  1 !
;   107          0433  1 ! Formal Parameters:
;   108          0434  1 !       input           Address of descriptor of input.
;   109          0435  1 !
;   110          0436  1 ! Implicit Inputs:
;   111          0437  1 !       global data
;   112          0438  1 !
;   113          0439  1 ! Implicit Outputs:
;   114          0440  1 !       global data
;   115          0441  1 !
;   116          0442  1 ! Returned Value:
;   117          0443  1 !       none
;   118          0444  1 !
;   119          0445  1 ! Side Effects:
;   120          0446  1 !
;   121          0447  1 !--
;   122          0448  1
;   123          0449  1
;   124          0450  2 global routine phn$kbd_route(input): novalue = begin
;   125          0451  2
;   126          0452  2 bind
;   127          0453  2         input_dsc = .input: descriptor;
;   128          0454  2
;   129          0455  2 own
;   130          0456  2         command_flag: byte initial(false);
;   131          0457  2
;   132          0458  2 own
;   133          0459  2         text_buf: vector[80,byte],              ! These variables are OWN due
;   134          0460  2         buf_i: long;                            ! up-level referencing.
;   135          0461  2
;   136          0462  2 local
;   137          0463  2         conversing: byte,
;   138          0464  2         char: byte,
;   139          0465  2         p: ref pub;
;   140          0466  2
;   141          0467  2 bind
;   142          0468  2         ctrl_w_cmd = describe(%string(%char(ctrl_u,ctrl_w,ret))),
;   143          0469  2         exit_cmd   = describe(%string(%char(ctrl_u),'EXIT',%char(ret))),
;   144          0470  2         hangup_cmd = describe(%string(%char(ctrl_u),'HANGUP',%char(ret)));
;   145          0471  2
;   146          0472  2
;   147          0473  2 ! The following routine is called when a collection of characters is ready
;   148          0474  2 ! to be queued to the appropriate steering message routine.  If the collection
;   149          0475  2 ! is non-empty, we queue a steering message.  We also clear the buffer for the
;   150          0476  2 ! next collection.
;   151          0477  2
;   152          0478  3 routine queue_collection: novalue = begin
;   153          0479  3
;   154          0480  3 local
;   155          0481  3         collection_dsc: descriptor;
```

```
;  156          0482  3
;  157          0483  4 if .buf_i gtru 0 then (
;  158          0484  4         collection_dsc[len] = .buf_i;
;  159          0485  4         collection_dsc[ptr] = text_buf;
;  160          0486  4         phn$queue_smb(if .command_flag then smb__cmd_parse else smb__talk,
;  161          0487  4                       collection_dsc);
;  162          0488  4         buf_i = 0;
;  163          0489  3 );
;  164          0490  3 return;
;  165          0491  2 end;


                                                        .TITLE  INPUT INPUT - Analyze and Act on Input
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                        0D  17  15  00000 P.AAB:  .ASCII  <21><23><13>                          ;
                                                    00003         .BLKB   1
                                          00000003  00004 P.AAA:  .LONG   3                                     ;
                                          00000000' 00008         .ADDRESS P.AAB                                ;
                                    0D  54  49  58  45  15  0000C P.AAD:  .ASCII  <21>\EXIT\<13>                 ;
                                                    00012         .BLKB   2
                                          00000006  00014 P.AAC:  .LONG   6                                     ;
                                          00000000' 00018         .ADDRESS P.AAD                                ;
                                50  55  47  4E  41  48  15  0001C P.AAF:  .ASCII  <21>\HANGUP\<13>               ;
                                          00000008  00024 P.AAE:  .LONG   8                                     ;
                                          00000000' 00028         .ADDRESS P.AAF                                ;

                                                        .PSECT  $OWN$,NOEXE,2

                                00  00000 COMMAND_FLAG:
                                                        .BYTE   0                                               ;
                                    00001               .BLKB   3
                                    00004 TEXT_BUF:
                                                        .BLKB   80
                                    00054 BUF_i:        .BLKB   4

                                          CTRL_W_CMD=           P.AAA
                                          EXIT_CMD=             P.AAC
                                          HANGUP_CMD=           P.AAE
                                                        .EXTRN  PHN$_OK, PHN$_ANSWERED
                                                        .EXTRN  PHN$_BUSYCALL, PHN$_CANCALL
                                                        .EXTRN  PHN$_CANTREACH, PHN$_CONFCALL
                                                        .EXTRN  PHN$_DEAD, PHN$_DECNETLINK
                                                        .EXTRN  PHN$_DIRCAN, PHN$_FACSCAN
                                                        .EXTRN  PHN$_HELPCAN, PHN$_HUNGUP
                                                        .EXTRN  PHN$_JUSTRANG, PHN$_LOGGEDOFF
                                                        .EXTRN  PHN$_REJECTED, PHN$_RING
                                                        .EXTRN  PHN$_REJECTJUNK
                                                        .EXTRN  PHN$_SENDINGMAIL
                                                        .EXTRN  PHN$_BADCMD, PHN$_BADHELP
                                                        .EXTRN  PHN$_BADMAILCMD
                                                        .EXTRN  PHN$_BADSMB, PHN$_BADSPEC
                                                        .EXTRN  PHN$_HELPMISSING
                                                        .EXTRN  PHN$_IVREDUNANS
                                                        .EXTRN  PHN$_IVREDUNCALL
```

```
                                .EXTRN  PHN$_LINKERROR, PHN$_NEEDUSER
                                .EXTRN  PHN$_NOCALL, PHN$_NOHOLDS
                                .EXTRN  PHN$_NOPORTS, PHN$_NOPRIV
                                .EXTRN  PHN$_NOPROC, PHN$_NOTCONV
                                .EXTRN  PHN$_ONLYNODE, PHN$_PHONEBUSY
                                .EXTRN  PHN$_REMOTEERROR
                                .EXTRN  PHN$_TARGTERM, PHN$_UNPLUGGED
                                .EXTRN  PHN$_BADTERM, PHN$_SHAREDMBX
                                .EXTRN  PHN$_INPUTTERM, PHN$GQ_NODE_NAME
                                .EXTRN  PHN$GQ_SWITCH_HOOK
                                .EXTRN  PHN$GL_VIEWPORT_SIZE
                                .EXTRN  PHN$GB_SCROLL, PHN$GQ_PUBHEAD
                                .EXTRN  PHN$GB_FLAGS, LIB$TPARSE
                                .EXTRN  PHN$ANSWER_CMD, PHN$BREAK_CALL
                                .EXTRN  PHN$CONVERSING, PHN$DIAL_CMD
                                .EXTRN  PHN$DIRECTORY_CMD
                                .EXTRN  PHN$EXIT_CMD, PHN$FACSIMILE_CMD
                                .EXTRN  PHN$FRESH_SCREEN
                                .EXTRN  PHN$HANGUP_CMD, PHN$HELP_CMD
                                .EXTRN  PHN$HOLD_CMD, PHN$INFORM
                                .EXTRN  PHN$MAIL_CMD, PHN$QUEUE_SMB
                                .EXTRN  PHN$REJECT_CMD, PHN$SHOW_TEXT
                                .EXTRN  PHN$TRANSCRIBE_CMD
                                .EXTRN  PHN$UNHOLD_CMD, LIB$_SYNTAXERR

                                .PSECT  $CODE$,NOWRT,2

                        0004 00000 QUEUE_COLLECTION:
                                .WORD   Save R2                                 ; 0478
                52      0000'   CF  9E 00002    MOVAB   BUF_I, R2              ;
                5E              08  C2 00007    SUBL2   #8, SP                 ;
                50              62  D0 0000A    MOVL    BUF_I, R0              ; 0483
                                1B  13 0000D    BEQL    3$                     ;
                6E              50  B0 0000F    MOVW    R0, COLLECTION_DSC     ; 0484
             04 AE          B0  A2  9E 00012    MOVAB   TEXT_BUF, COLLECTION_DSC+4 ; 0485
                5E              DD  00017       PUSHL   SP                     ; 0486
                04          AC  A2  E9 00019    BLBC    COMMAND_FLAG, 1$       ;
                                03  DD 0001D    PUSHL   #3                     ;
                                02  11 0001F    BRB     2$                     ;
                                04  DD 00021 1$: PUSHL   #4                    ;
            0000G CF            02  FB 00023 2$: CALLS   #2, PHN$QUEUE_SMB     ;
                                62  D4 00028    CLRL    BUF_I                  ; 0488
                                    04 0002A 3$: RET                           ; 0491

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  167          0492  2 ! The first thing we have to do is check to see what state the screen is in.
;  168          0493  2 ! If the scroller flag is set, some scrolling-type command (e.g., HELP)
;  169          0494  2 ! is in progress.  We need to reset the flag to cancel the command,
;  170          0495  2 ! and refresh the screen.
;  171          0496  2
;  172          0497  3 if .phn$gv_scroller then (
;  173          0498  3         phn$gv_scroller = false;
;  174          0499  3         if ch$find_ch(.input_dsc[len],.input_dsc[ptr],ctrl_w) eqla 0 then
;  175          0500  3                 phn$fresh_screen(true);
;  176          0501  2 );
;  177          0502  2
;  178          0503  2 ! Next thing we do is check to see if a message is sitting on the
;  179          0504  2 ! message line.  If so, we want to clear it out.
;  180          0505  2
;  181          0506  3 if .phn$gv_message then (
;  182          0507  3         phn$gv_message = false;
;  183          0508  3         phn$inform();
;  184          0509  2 );
;  185          0510  2
;  186          0511  2 ! Another thing we need to check for is whether the user is currently
;  187          0512  2 ! ringing someone's phone.  If so, we cancel the call immediately.
;  188          0513  2
;  189          0514  2 p = .phn$gq_pubhead[0];
;  190          0515  2 if .p[pub_v_calling] then
;  191          0516  2         phn$break_call();
;  192          0517  2
;  193          0518  2 ! Finally, we have to see if a facsimile operation is in progress.
;  194          0519  2 ! If so, we cancel it by turning off the flag.
;  195          0520  2
;  196          0521  2 phn$gv_facsimile = false;
```

```
;  198          0522  2 ! Now we are going to look at the input typed by the user.  We will split
;  199          0523  2 ! it up into conversation text and command text.
;  200          0524  2
;  201          0525  2 conversing = phn$conversing();
;  202          0526  2 buf_i = 0;
;  203          0527  2
;  204          0528  3 while dec (input_dsc[len]) geq 0 do (
;  205          0529  3
;  206          0530  3         char = ch$rchar_a(input_dsc[ptr]);
;  207          0531  3
;  208          0532  3         command_flag = .command_flag or (not .conversing);
;  209          0533  3         if .command_flag then
;  210          0534  3
;  211          0535  3                 ! We are in the middle of a command.  Process each character:
;  212          0536  3                 !       return          Done with this command, back to talking
;  213          0537  3                 !       switch hook     Done with this command, start another.
;  214          0538  3                 !       CTRL/W          Flush current command & refresh screen.
;  215          0539  3                 !       CTRL/Z          Flush current command & force EXIT.
;  216          0540  3                 !       others          Add to current command.
;  217          0541  3
;  218          0542  3                 select .char of set
;  219          0543  4                 [ret]:          (text_buf[.buf_i] = ret;
;  220          0544  4                                 inc(buf_i);
;  221          0545  4                                 queue_collection();
;  222          0546  3                                 command_flag = false;);
;  223          0547  3
;  224          0548  3                 [ch$rchar(.phn$g_switch_hook[ptr])]:
;  225          0549  4                                 (text_buf[.buf_i] = ret;
;  226          0550  4                                 inc(buf_i);
;  227          0551  3                                 queue_collection(););
;  228          0552  3
;  229          0553  4                 [ctrl_w]:       (queue_collection();
;  230          0554  4                                 phn$queue_smb(smb__cmd_parse,ctrl_w_cmd);
;  231          0555  3                                 command_flag = false;);
;  232          0556  3
;  233          0557  4                 [ctrl_z]:       (queue_collection();
;  234          0558  4                                 phn$queue_smb(smb__cmd_parse,exit_cmd);
;  235          0559  3                                 command_flag = false;);
;  236          0560  3
;  237          0561  4                 [otherwise]:    (text_buf[.buf_i] = .char;
;  238          0562  3                                 inc(buf_i););
;  239          0563  3                 tes
```

```
;  241          0564  3            else
;  242          0565  3
;  243          0566  3                 ! The user is entering some conversation text.  Process each
;  244          0567  3                 ! character as follows:
;  245          0568  3                 !       switch hook     Suspend talking, start a new command.
;  246          0569  3                 !       CTRL/W          Suspend talking, refresh screen.
;  247          0570  3                 !       CTRL/Z          Suspend talking, force HANGUP command.
;  248          0571  3                 !       others          Add to conversation text.
;  249          0572  3
;  250          0573  3                 selectoneu .char of set
;  251          0574  3                 [ch$rchar(.phn$gq_switch_hook[ptr])]:
;  252          0575  4                                 (queue_collection();
;  253          0576  3                                 command_flag = true;);
;  254          0577  3
;  255          0578  4                 [ctrl_w]:       (queue_collection();
;  256          0579  3                                 phn$queue_smb(smb__cmd_parse,ctrl_w_cmd),);
;  257          0580  3
;  258          0581  4                 [ctrl_z]:       (queue_collection();
;  259          0582  3                                 phn$queue_smb(smb__cmd_parse,hangup_cmd););
;  260          0583  3
;  261          0584  4                 [otherwise]:    (text_buf[.buf_i] = .char;
;  262          0585  3                                 inc(buf_i););
;  263          0586  3                 tes;
;  264          0587  3
;  265          0588  2 );
;  266          0589  2
;  267          0590  2 ! Make sure we queue any final collection we were in the process of building.
;  268          0591  2
;  269          0592  2 queue_collection();
;  270          0593  2 return;
;  271          0594  2
;  272          0595  1 end;
```

```
                                    01FC 00000          .ENTRY  PHN$KBD_ROUTE, Save R2,R3,R4,R5,R6,R7,R8    ; 0450
                              58    0000'  CF   9E 00002  MOVAB   CTRL_W_CMD, R8                              ;
                              57    0000G  CF   9E 00007  MOVAB   PHN$GB_FLAGS, R7                           ;
                              56      C6   AF   9E 0000C  MOVAB   QUEUE_COLLECTION, R6                       ;
                              55    0000'  CF   9E 00010  MOVAB   BUF_I, R5                                  ;
                              53      04   AC   D0 00015  MOVL    INPUT, R3                                  ; 0453
                    17        67           01   E1 00019  BBC     #1, PHN$GB_FLAGS, 2$                       ; 0497
                              67           02   8A 0001D  BICB2   #2, PHN$GB_FLAGS                          ; 0498
                 04   B3      63           17   3A 00020  LOCC    #23, (R3), @4(R3)                          ; 0499
                                           02   12 00025  BNEQ    1$                                         ;
                                           51   D4 00027  CLRL    R1                                         ;
                                           51   D5 00029 1$:  TSTL    R1                                         ;
                                           07   12 0002B  BNEQ    2$                                         ;
                                           01   DD 0002D  PUSHL   #1                                         ; 0500
                         0000G CF          01   FB 0002F  CALLS   #1, PHN$FRESH_SCREEN                       ;
                              08           67   E9 00034 2$:  BLBC    PHN$GB_FLAGS, 3$                           ; 0506
                              67           01   8A 00037  BICB2   #1, PHN$GB_FLAGS                          ; 0507
                         0000G CF          00   FB 0003A  CALLS   #0, PHN$INFORM                            ; 0508
                              50    0000G  CF   D0 0003F 3$:  MOVL    PHN$GQ_PUBHEAD, P                          ; 0514
                    05   00F0  C0          03   E1 00044  BBC     #3, 240(P), 4$                             ; 0515
```

```
                0000G  CF              00  FB 0004A          CALLS   #0, PHN$BREAK_CALL                          ; 0516
                       67              08  8A 0004F 4$:      BICB2   #8, PHN$GB_FLAGS                          ; 0521
                0000G  CF              00  FB 00052          CALLS   #0, PHN$CONVERSING                        ; 0525
                       54              50  90 00057          MOVB    R0, CONVERSING                            ;
                                       65  D4 0005A          CLRL    BUF_I                                     ; 0526
                       50              63  3C 0005C 5$:      MOVZWL  (R3), R0                                  ; 0528
                                       50  D7 0005F          DECL    R0                                        ;
                       63              50  B0 00061          MOVW    R0, (R3)                                  ;
                                       50  D5 00064          TSTL    R0                                        ;
                                       03  18 00066          BGEQ    6$                                        ;
                                     00A5  31 00068          BRW     18$                                       ;
                       52        04    B3  90 0006B 6$:      MOVB    @4(R3), CHAR                              ; 0530
                                 04    A3  D6 0006F          INCL    4(R3)                                     ;
                       50              54  9A 00072          MOVZBL  CONVERSING, R0                            ; 0532
                       51        AC    A5  9A 00075          MOVZBL  COMMAND_FLAG, R1                          ;
                       50              51  CA 00079          BICL2   R1, R0                                    ;
                    AC 50              50  92 0007C          MCOMB   R0, COMMAND_FLAG                          ;
                       4F        AC    A5  E9 00080          BLBC    COMMAND_FLAG, 13$                         ; 0533
                       0D              52  91 00084          CMPB    CHAR, #13                                 ; 0543
                                       10  12 00087          BNEQ    7$                                        ;
                       50        B0    A5  9E 00089          MOVAB   TEXT_BUF, R0                              ;
                    00 B540            0D  90 0008D          MOVB    #13, @BUF_I[R0]                           ;
                                       65  D6 00092          INCL    BUF_I                                     ; 0544
                       66              00  FB 00094          CALLS   #0, QUEUE_COLLECTION                      ; 0545
                                       35  11 00097          BRB     12$                                       ; 0546
                0000G  DF              52  91 00099 7$:      CMPB    CHAR, @PHN$GQ_SWITCH_HOOK+4               ; 0548
                                       10  12 0009E          BNEQ    9$                                        ;
                       50        B0    A5  9E 000A0          MOVAB   TEXT_BUF, R0                              ; 0549
                    00 B540            0D  90 000A4          MOVB    #13, @BUF_I[R0]                           ;
                                       65  D6 000A9          INCL    BUF_I                                     ; 0550
                       66              00  FB 000AB          CALLS   #0, QUEUE_COLLECTION                      ; 0551
                                       AC  11 000AE 8$:      BRB     5$                                        ; 0542
                       17              52  91 000B0 9$:      CMPB    CHAR, #23                                 ; 0553
                                       07  12 000B3          BNEQ    10$                                       ;
                       66              00  FB 000B5          CALLS   #0, QUEUE_COLLECTION                      ;
                                       58  DD 000B8          PUSHL   R8                                        ; 0554
                                       0B  11 000BA          BRB     11$                                       ;
                       1A              52  91 000BC 10$:     CMPB    CHAR, #26                                 ; 0557
                                       42  12 000BF          BNEQ    17$                                       ;
                       66              00  FB 000C1          CALLS   #0, QUEUE_COLLECTION                      ;
                                 10    A8  9F 000C4          PUSHAB  EXIT_CMD                                  ; 0558
                                       03  DD 000C7 11$:     PUSHL   #3                                        ;
                0000G  CF              02  FB 000C9          CALLS   #2, PHN$QUEUE_SMB                         ;
                                 AC    A5  94 000CE 12$:     CLRB    COMMAND_FLAG                              ; 0559
                                       89  11 000D1          BRB     5$                                        ; 0542
                0000G  DF              52  91 000D3 13$:     CMPB    CHAR, @PHN$GQ_SWITCH_HOOK+4               ; 0574
                                       09  12 000D8          BNEQ    14$                                       ;
                       66              00  FB 000DA          CALLS   #0, QUEUE_COLLECTION                      ; 0575
                    AC 01              01  90 000DD          MOVB    #1, COMMAND_FLAG                          ; 0576
                                       CB  11 000E1          BRB     8$                                        ; 0573
                       17              52  91 000E3 14$:     CMPB    CHAR, #23                                 ; 0578
                                       07  12 000E6          BNEQ    15$                                       ;
                       66              00  FB 000E8          CALLS   #0, QUEUE_COLLECTION                      ;
                                       58  DD 000EB          PUSHL   R8                                        ; 0579
                                       0B  11 000ED          BRB     16$                                       ;
                       1A              52  91 000EF 15$:     CMPB    CHAR, #26                                 ; 0581
                                       0F  12 000F2          BNEQ    17$                                       ;
```

```
                                66              00  FB 000F4          CALLS   #0, QUEUE_COLLECTION                    ;
                                          20    A8  9F 000F7          PUSHAB  HANGUP_CMD                            ; 0582
                                                03  DD 000FA 16$:     PUSHL   #3                                    ;
                          0000G CF              02  FB 000FC          CALLS   #2, PHN$QUEUE_SMB                     ;
                                                AB  11 00101          BRB     8$                                    ; 0573
                                50        B0    A5  9E 00103 17$:     MOVAB   TEXT_BUF, R0                          ; 0584
                             00 B540            52  90 00107          MOVB    CHAR, @BUF_I[R0]                      ;
                                                65  D6 0010C          INCL    BUF_I                                 ; 0585
                                                9E  11 0010E          BRB     8$                                    ; 0528
                                66              00  FB 00110 18$:     CALLS   #0, QUEUE_COLLECTION                  ; 0592
                                                04  00113           RET                                           ; 0595

; Routine Size:  276 bytes,    Routine Base:  $CODE$ + 002B
```

```
;  274          0596  1 %sbttl 'PHN$CMD_PARSE - Parse a Command'
;  275          0597  1 !++
;  276          0598  1 ! Functional Description:
;  277          0599  1 !       This steering message routine is invoked when we have part
;  278          0600  1 !       or all of a command from PHN$KBD_ROUTE.  We collect characters
;  279          0601  1 !       until we have a complete command, and then we parse and execute it.
;  280          0602  1 !
;  281          0603  1 ! Formal Parameters:
;  282          0604  1 !       cmd_text        Part or all of a command.  Never a null string and
;  283          0605  1 !                       never parts of two commands.
;  284          0606  1 !
;  285          0607  1 ! Implicit Inputs:
;  286          0608  1 !       global data
;  287          0609  1 !
;  288          0610  1 ! Implicit Outputs:
;  289          0611  1 !       global data
;  290          0612  1 !
;  291          0613  1 ! Returned Value:
;  292          0614  1 !       none
;  293          0615  1 !
;  294          0616  1 ! Side Effects:
;  295          0617  1 !
;  296          0618  1 !--
;  297          0619  1
;  298          0620  1
;  299          0621  2 global routine phn$cmd_parse(cmd_text): novalue = begin
;  300          0622  2
;  301          0623  2 bind
;  302          0624  2         text_dsc = .cmd_text: descriptor;
;  303          0625  2
;  304          0626  2 own
;  305          0627  2         command_buf: vector[80,byte],
;  306          0628  2         buf_i: long initial(0),
;  307          0629  2
;  308          0630  2         tparse_block: block[tpa$k_length0,byte]
;  309          0631  2                 initial(tpa$k_count0,
;  310          0632  2                         tpa$m_blanks + tpa$m_abbrev),
;  311          0633  2         command_proc: long;
;  312          0634  2
;  313          0635  2 local
;  314          0636  2         status: long,
;  315          0637  2         command_complete: byte,
;  316          0638  2         char: byte,
;  317          0639  2         in_quotes: byte, i: long;
```

```
;  319          0640  2 ! The following is the parsing table used to analyze a command and
;  320          0641  2 ! determine which command routine to call.
;  321          0642  2 ! The TRANSCRIBE command has been more or less temporarily removed.
;  322          0643  2
;  323          0644  2 $init_state(command_state,command_key);
;  324          0645  2
;  325        P 0646  2 $state  (,
;  326        P 0647  2                 (tpa$_blank),
;  327        P 0648  2                 (tpa$_lambda)
;  328          0649  2          );
;  329          0650  2
;  330        P 0651  2 $state  (,
;  331        P 0652  2                 (tpa$_eos,tpa$_exit),
;  332        P 0653  2                 (ctrl_w,        noargs),
;  333        P 0654  2                 ('ANSWER',      noargs,,phn$answer_cmd,         command_proc),
;  334        P 0655  2                 ('DIAL',          args,,phn$dial_cmd,           command_proc),
;  335        P 0656  2                 ('DIRECTORY',     args,,phn$directory_cmd,      command_proc),
;  336        P 0657  2                 ('EXIT',        noargs,,phn$exit_cmd,           command_proc),
;  337        P 0658  2                 ('FACSIMILE',     args,,phn$facsimile_cmd,      command_proc),
;  338        P 0659  2                 ('HANGUP',      noargs,,phn$hangup_cmd,         command_proc),
;  339        P 0660  2                 ('HELP',          args,,phn$help_cmd,           command_proc),
;  340        P 0661  2                 ('HOLD',        noargs,,phn$hold_cmd,           command_proc),
;  341        P 0662  2                 ('MAIL',          args,,phn$mail_cmd,           command_proc),
;  342        P 0663  2                 ('PHONE',         args,,phn$dial_cmd,           command_proc),
;  343        P 0664  2                 ('REJECT',        args,,phn$reject_cmd,         command_proc),
;  344        P 0665  2 !               ('TRANSCRIBE',    args,,phn$transcribe_cmd,     command_proc),
;  345        P 0666  2                 ('UNHOLD',      noargs,,phn$unhold_cmd,         command_proc),
;  346        P 0667  2                 (tpa$_lambda,     args,,phn$dial_cmd,           command_proc)
;  347          0668  2          );
;  348          0669  2
;  349        P 0670  2 $state  (noargs,
;  350        P 0671  2                 (tpa$_blank),
;  351        P 0672  2                 (tpa$_lambda)
;  352          0673  2          );
;  353        P 0674  2 $state  (,
;  354        P 0675  2                 (tpa$_eos,tpa$_exit)
;  355          0676  2          );
;  356          0677  2
;  357        P 0678  2 $state  (args,
;  358        P 0679  2                 (tpa$_blank,tpa$_exit),
;  359        P 0680  2                 (tpa$_lambda,tpa$_exit)
;  360          0681  2          );
```

```
;  362     0682  2 ! We begin by displaying the command fragment on the screen.
;  363     0683  2
;  364     0684  2 phn$show_text(0,text_dsc);
;  365     0685  2
;  366     0686  2 ! We look at each character that has been passed to us.  Note that a
;  367     0687  2 ! similar analysis has to be done in PHN$SHOW_TEXT.  There are the
;  368     0688  2 ! following cases:
;  369     0689  2 !       normal          Add character to command.
;  370     0690  2 !       return          End of the command.  Parse & execute it.
;  371     0691  2 !       delete          Delete last character in command.
;  372     0692  2 !       CTRL/U          Flush entire command and start over.
;  373     0693  2
;  374     0694  2 command_complete = false;
;  375     0695  3 while dec (text_dsc[len]) geq 0 do (
;  376     0696  3
;  377     0697  3         char = ch$rchar_a(text_dsc[ptr]);
;  378     0698  3
;  379     0699  3         selectoneu .char of set
;  380     0700  3         [%x'20' to %x'7e',
;  381     0701  3          %x'80' to %x'ff']:
;  382     0702  4                         (command_buf[.buf_i] = .char;
;  383     0703  3                         buf_i = minu(.buf_i+1,79););
;  384     0704  3
;  385     0705  3         [ret]:          command_complete = true;
;  386     0706  3
;  387     0707  3         [delete]:       buf_i = max(.buf_i-1, 0);
;  388     0708  3
;  389     0709  3         [ctrl_u]:       buf_i = 0;
;  390     0710  3         tes;
;  391     0711  3
;  392     0712  2 );
;  393     0713  2
;  394     0714  2 ! If we have a complete command now, we can parse it.  All we are interested
;  395     0715  2 ! in here is the command verb; all arguments will be passed on to the
;  396     0716  2 ! command processing routine.
;  397     0717  2
;  398     0718  2 if not .command_complete then
;  399     0719  2         return;
;  400     0720  2
;  401     0721  2 ! We need to uppercase the command for comparison purposes.  However, any
;  402     0722  2 ! quoted strings should be left alone.
;  403     0723  2
;  404     0724  2 in_quotes = false;
;  405     0725  3 incr i from 0 to .buf_i-1 do (
;  406     0726  3         if not .in_quotes and
;  407     0727  3            .command_buf[.i] gegu 'a' and .command_buf[.i] legu 'z' then
;  408     0728  3                 command_buf[.i] = .command_buf[.i] and %b'11011111';
;  409     0729  3         in_quotes = .in_quotes xor (.command_buf[.i] eqlu '"');
;  410     0730  2 );
;  411     0731  2
;  412     0732  2 tparse_block[tpa$l_stringcnt] = .buf_i;
;  413     0733  2 tparse_block[tpa$l_stringptr] = command_buf;
;  414     0734  2 command_proc = 0;
;  415     0735  2 status = lib$tparse(tparse_block,command_state,command_key);
;  416     0736  2
;  417     0737  2 ! If we didn't get a syntax error, we can call the command procedure, passing
;  418     0738  2 ! it a descriptor of the command arguments (which it can clobber).
```

```
;  419          0739  2
;  420          0740  2 if .status eqlu lib$_syntaxerr then
;  421          0741  2         phn$inform(phn$_badcmd)
;  422          0742  3 else (
;  423          0743  3         check (.status);
;  424          0744  3         if .command_proc neqa 0 then
;  425          0745  3                 (.command_proc) (tparse_block[tpa$l_stringcnt]);
;  426          0746  2 );
;  427          0747  2
;  428          0748  2 ! All done.  Clear out command line on screen and reset our buffer.
;  429          0749  2
;  430          0750  2 phn$show_text(0,describe(%char(ctrl_u)));
;  431          0751  2 buf_i = 0;
;  432          0752  2 return;
;  433          0753  2
;  434          0754  1 end;


                                                .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                        00000 ;TPA$KEYSTO
                                        U.9:    .BLKB   0
                    52  45  57  53  4E  41  00000 ;TPA$KEYST
                                        U.11:   .ASCII  \ANSWER\                                ;
                                    FF  00006           .BYTE   -1                              ;
                                        00007 ;TPA$KEYSTO
                                        U.16:   .BLKB   0
                            4C  41  49  44  00007 ;TPA$KEYST
                                        U.18:   .ASCII  \DIAL\                                  ;
                                    FF  0000B           .BYTE   -1                              ;
                                        0000C ;TPA$KEYSTO
                                        U.24:   .BLKB   0
                59  52  4F  54  43  45  52  49  44  0000C ;TPA$KEYST
                                        U.26:   .ASCII  \DIRECTORY\                             ;
                                    FF  00015           .BYTE   -1                              ;
                                        00016 ;TPA$KEYSTO
                                        U.31:   .BLKB   0
                                54  49  58  45  00016 ;TPA$KEYST
                                        U.33:   .ASCII  \EXIT\                                  ;
                                    FF  0001A           .BYTE   -1                              ;
                                        0001B ;TPA$KEYSTO
                                        U.38:   .BLKB   0
                45  4C  49  4D  49  53  43  41  46  0001B ;TPA$KEYST
                                        U.40:   .ASCII  \FACSIMILE\                             ;
                                    FF  00024           .BYTE   -1                              ;
                                        00025 ;TPA$KEYSTO
                                        U.45:   .BLKB   0
                        50  55  47  4E  41  48  00025 ;TPA$KEYST
                                        U.47:   .ASCII  \HANGUP\                                ;
                                    FF  0002B           .BYTE   -1                              ;
                                        0002C ;TPA$KEYSTO
                                        U.52:   .BLKB   0
                            50  4C  45  48  0002C ;TPA$KEYST
                                        U.54:   .ASCII  \HELP\                                  ;
                                    FF  00030           .BYTE   -1                              ;
                                        00031 ;TPA$KEYSTO
```

```
                                                U.59:    .BLKB   0
                                 44 4C 4F 48 00031 ;TPA$KEYST
                                                U.61:    .ASCII  \HOLD\                  ;
                                    FF 00035            .BYTE   -1                      ;
                                       00036 ;TPA$KEYSTO
                                                U.66:    .BLKB   0
                                 4C 49 41 4D 00036 ;TPA$KEYST
                                                U.68:    .ASCII  \MAIL\                  ;
                                    FF 0003A            .BYTE   -1                      ;
                                       0003B ;TPA$KEYSTO
                                                U.73:    .BLKB   0
                              45 4E 4F 48 50 0003B ;TPA$KEYST
                                                U.75:    .ASCII  \PHONE\                 ;
                                    FF 00040            .BYTE   -1                      ;
                                       00041 ;TPA$KEYSTO
                                                U.80:    .BLKB   0
                           54 43 45 4A 45 52 00041 ;TPA$KEYST
                                                U.82:    .ASCII  \REJECT\                ;
                                    FF 00047            .BYTE   -1                      ;
                                       00048 ;TPA$KEYSTO
                                                U.87:    .BLKB   0
                           44 4C 4F 48 4E 55 00048 ;TPA$KEYST
                                                U.89:    .ASCII  \UNHOLD\                ;
                                    FF 0004E            .BYTE   -1                      ;
                                    FF 0004F ;TPA$KEYFILL
                                                U.98:    .BYTE   -1                      ;

                                                        .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                                       00000 COMMAND_STATE::
                                                        .BLKB   0
                                 01F2  00000 ;TPA$TYPE
                                                U.2:     .WORD   498                     ;
                                 05F6  00002 ;TPA$TYPE
                                                U.3:     .WORD   1526                    ;
                                 11F7  00004 ;TPA$TYPE
                                                U.4:     .WORD   4599                    ;
                                 FFFF  00006 ;TPA$TARGET
                                                U.5:     .WORD   -1                      ;
                                 1017  00008 ;TPA$TYPE
                                                U.6:     .WORD   4119                    ;
                                 0000* 0000A ;TPA$TARGET
                                                U.8:     .WORD   <<U.7-U.8>-2>           ;
                                 7100  0000C ;TPA$TYPE
                                                U.12:    .WORD   28928                   ;
                           00000000* 0000E ;TPA$ADDR
                                                U.13:    .LONG   <<COMMAND_PROC-U.13>-4> ;
                           00000000G 00012 ;TPA$MASK
                                                U.14:    .ADDRESS PHN$ANSWER_CMD         ;
                                 0000* 00016 ;TPA$TARGET
                                                U.15:    .WORD   <<U.7-U.15>-2>          ;
                                 7101  00018 ;TPA$TYPE
                                                U.19:    .WORD   28929                   ;
                           00000000* 0001A ;TPA$ADDR
                                                U.20:    .LONG   <<COMMAND_PROC-U.20>-4> ;
                           00000000G 0001E ;TPA$MASK
                                                U.21:    .ADDRESS PHN$DIAL_CMD           ;
```

```
                0000* 00022 ;TPA$TARGET
                            U.23:    .WORD    <<U.22-U.23>-2>                    ;
                7102  00024 ;TPA$TYPE
                            U.27:    .WORD    28930                              ;
            00000000* 00026 ;TPA$ADDR
                            U.28:    .LONG    <<COMMAND_PROC-U.28>-4>            ;
            00000000G 0002A ;TPA$MASK
                            U.29:    .ADDRESS PHN$DIRECTORY_CMD                  ;
                0000* 0002E ;TPA$TARGET
                            U.30:    .WORD    <<U.22-U.30>-2>                    ;
                7103  00030 ;TPA$TYPE
                            U.34:    .WORD    28931                              ;
            00000000* 00032 ;TPA$ADDR
                            U.35:    .LONG    <<COMMAND_PROC-U.35>-4>            ;
            00000000G 00036 ;TPA$MASK
                            U.36:    .ADDRESS PHN$EXIT_CMD                       ;
                0000* 0003A ;TPA$TARGET
                            U.37:    .WORD    <<U.7-U.37>-2>                     ;
                7104  0003C ;TPA$TYPE
                            U.41:    .WORD    28932                              ;
            00000000* 0003E ;TPA$ADDR
                            U.42:    .LONG    <<COMMAND_PROC-U.42>-4>            ;
            00000000G 00042 ;TPA$MASK
                            U.43:    .ADDRESS PHN$FACSIMILE_CMD                  ;
                0000* 00046 ;TPA$TARGET
                            U.44:    .WORD    <<U.22-U.44>-2>                    ;
                7105  00048 ;TPA$TYPE
                            U.48:    .WORD    28933                              ;
            00000000* 0004A ;TPA$ADDR
                            U.49:    .LONG    <<COMMAND_PROC-U.49>-4>            ;
            00000000G 0004E ;TPA$MASK
                            U.50:    .ADDRESS PHN$HANGUP_CMD                     ;
                0000* 00052 ;TPA$TARGET
                            U.51:    .WORD    <<U.7-U.51>-2>                     ;
                7106  00054 ;TPA$TYPE
                            U.55:    .WORD    28934                              ;
            00000000* 00056 ;TPA$ADDR
                            U.56:    .LONG    <<COMMAND_PROC-U.56>-4>            ;
            00000000G 0005A ;TPA$MASK
                            U.57:    .ADDRESS PHN$HELP_CMD                       ;
                0000* 0005E ;TPA$TARGET
                            U.58:    .WORD    <<U.22-U.58>-2>                    ;
                7107  00060 ;TPA$TYPE
                            U.62:    .WORD    28935                              ;
            00000000* 00062 ;TPA$ADDR
                            U.63:    .LONG    <<COMMAND_PROC-U.63>-4>            ;
            00000000G 00066 ;TPA$MASK
                            U.64:    .ADDRESS PHN$HOLD_CMD                       ;
                0000* 0006A ;TPA$TARGET
                            U.65:    .WORD    <<U.7-U.65>-2>                     ;
                7108  0006C ;TPA$TYPE
                            U.69:    .WORD    28936                              ;
            00000000* 0006E ;TPA$ADDR
                            U.70:    .LONG    <<COMMAND_PROC-U.70>-4>            ;
            00000000G 00072 ;TPA$MASK
                            U.71:    .ADDRESS PHN$MAIL_CMD                       ;
                0000* 00076 ;TPA$TARGET
```

```
                        U.72:   .WORD   <<U.22-U.72>-2>                 ;
    7109  00078 ;TPA$TYPE
                        U.76:   .WORD   28937                           ;
00000000* 0007A ;TPA$ADDR
                        U.77:   .LONG   <<COMMAND_PROC-U.77>-4>         ;
00000000G 0007E ;TPA$MASK
                        U.78:   .ADDRESS PHN$DIAL_CMD                   ;
    0000* 00082 ;TPA$TARGET
                        U.79:   .WORD   <<U.22-U.79>-2>                 ;
    710A  00084 ;TPA$TYPE
                        U.83:   .WORD   28938                           ;
00000000* 00086 ;TPA$ADDR
                        U.84:   .LONG   <<COMMAND_PROC-U.84>-4>         ;
00000000G 0008A ;TPA$MASK
                        U.85:   .ADDRESS PHN$REJECT_CMD                 ;
    0000* 0008E ;TPA$TARGET
                        U.86:   .WORD   <<U.22-U.86>-2>                 ;
    710B  00090 ;TPA$TYPE
                        U.90:   .WORD   28939                           ;
00000000* 00092 ;TPA$ADDR
                        U.91:   .LONG   <<COMMAND_PROC-U.91>-4>         ;
00000000G 00096 ;TPA$MASK
                        U.92:   .ADDRESS PHN$UNHOLD_CMD                 ;
    0000* 0009A ;TPA$TARGET
                        U.93:   .WORD   <<U.7-U.93>-2>                  ;
    75F6  0009C ;TPA$TYPE
                        U.94:   .WORD   30198                           ;
00000000* 0009E ;TPA$ADDR
                        U.95:   .LONG   <<COMMAND_PROC-U.95>-4>         ;
00000000G 000A2 ;TPA$MASK
                        U.96:   .ADDRESS PHN$DIAL_CMD                   ;
    0000* 000A6 ;TPA$TARGET
                        U.97:   .WORD   <<U.22-U.97>-2>                 ;
          000A8 ;NOARGS
                        U.7:    .BLKB   0
    01F2  000A8 ;TPA$TYPE
                        U.99:   .WORD   498                             ;
    05F6  000AA ;TPA$TYPE
                        U.100:  .WORD   1526                            ;
    15F7  000AC ;TPA$TYPE
                        U.101:  .WORD   5623                            ;
    FFFF  000AE ;TPA$TARGET
                        U.102:  .WORD   -1                              ;
          000B0 ;ARGS
                        U.22:   .BLKB   0
    11F2  000B0 ;TPA$TYPE
                        U.103:  .WORD   4594                            ;
    FFFF  000B2 ;TPA$TARGET
                        U.104:  .WORD   -1                              ;
    15F6  000B4 ;TPA$TYPE
                        U.105:  .WORD   5622                            ;
    FFFF  000B6 ;TPA$TARGET
                        U.106:  .WORD   -1                              ;

                                .PSECT  _LIB$KEY0$,NOWRT,  SHR,  PIC,1

          00000 COMMAND_KEY::
```

```
                                .BLKB   0
                00000 ;TPA$KEY0
                        U.1:    .BLKB   0
           0000* 00000 ;TPA$KEY
                        U.10:   .WORD   <U.9-U.1>                               ;
           0000* 00002 ;TPA$KEY
                        U.17:   .WORD   <U.16-U.1>                              ;
           0000* 00004 ;TPA$KEY
                        U.25:   .WORD   <U.24-U.1>                              ;
           0000* 00006 ;TPA$KEY
                        U.32:   .WORD   <U.31-U.1>                              ;
           0000* 00008 ;TPA$KEY
                        U.39:   .WORD   <U.38-U.1>                              ;
           0000* 0000A ;TPA$KEY
                        U.46:   .WORD   <U.45-U.1>                              ;
           0000* 0000C ;TPA$KEY
                        U.53:   .WORD   <U.52-U.1>                              ;
           0000* 0000E ;TPA$KEY
                        U.60:   .WORD   <U.59-U.1>                              ;
           0000* 00010 ;TPA$KEY
                        U.67:   .WORD   <U.66-U.1>                              ;
           0000* 00012 ;TPA$KEY
                        U.74:   .WORD   <U.73-U.1>                              ;
           0000* 00014 ;TPA$KEY
                        U.81:   .WORD   <U.80-U.1>                              ;
           0000* 00016 ;TPA$KEY
                        U.88:   .WORD   <U.87-U.1>                              ;

                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                     15 0002C P.AAH:    .ASCII  <21>                            ;
                        0002D           .BLKB   3
                00000001 00030 P.AAG:   .LONG   1                               ;
                00000000' 00034         .ADDRESS P.AAH                          ;

                                .PSECT  $OWN$,NOEXE,2

                        00058 COMMAND_BUF:
                                .BLKB   80
                00000000 000A8 BUF_I:   .LONG   0                               ;
          00000003 00000008 000AC TPARSE_BLOCK:
                                .LONG   8, 3                                    ;
                        000B4           .BLKB   28
                        000D0 COMMAND_PROC:
                                .BLKB   4


                                .PSECT  $CODE$,NOWRT,2

                           001C 00000   .ENTRY  PHN$CMD_PARSE, Save R2,R3,R4    ; 0621
                54    0000'  CF   9E 00002   MOVAB  BUF_I, R4                   ;
                52      04   AC   D0 00007   MOVL   CMD_TEXT, R2                ; 0624
                             52   DD 0000B   PUSHL  R2                          ; 0684
                             7E   D4 0000D   CLRL   -(SP)                       ;
          0000G CF           02   FB 0000F   CALLS  #2, PHN$SHOW_TEXT          ;
                             53   94 00014   CLRB   COMMAND_COMPLETE            ; 0694
```

```
                                50              62  3C 00016 1$:     MOVZWL  (R2), R0                        ; 0695
                                                50  D7 00019         DECL    R0                              ;
                                62              50  B0 0001B         MOVW    R0, (R2)                        ;
                                                50  D5 0001E         TSTL    R0                              ;
                                                5A  19 00020         BLSS    8$                              ;
                                51          04  B2  90 00022         MOVB    @4(R2), CHAR                    ; 0697
                                            04  A2  D6 00026         INCL    4(R2)                           ;
                                20              51  91 00029         CMPB    CHAR, #32                       ; 0700
                                                06  1F 0002C         BLSSU   2$                              ;
                            7E  8F              51  91 0002E         CMPB    CHAR, #126                      ;
                                                06  1B 00032         BLEQU   3$                              ;
                            80  8F              51  91 00034 2$:     CMPB    CHAR, #128                      ;
                                                1C  1F 00038         BLSSU   4$                              ;
                                50          B0  A4  9E 0003A 3$:     MOVAB   COMMAND_BUF, R0                 ; 0702
                            00  B440            51  90 0003E         MOVB    CHAR, @BUF_I[R0]                ;
                    50              64          01  C1 00043         ADDL3   #1, BUF_I, R0                   ; 0703
                0000004F    8F                  50  D1 00047         CMPL    R0, #79                         ;
                                                1E  1B 0004E         BLEQU   6$                              ;
                                50          4F  8F  9A 00050         MOVZBL  #79, R0                         ;
                                                18  11 00054         BRB     6$                              ;
                                0D              51  91 00056 4$:     CMPB    CHAR, #13                       ; 0705
                                                05  12 00059         BNEQ    5$                              ;
                                53              01  90 0005B         MOVB    #1, COMMAND_COMPLETE            ;
                                                B6  11 0005E         BRB     1$                              ;
                            7F  8F              51  91 00060 5$:     CMPB    CHAR, #127                      ; 0707
                                                0D  12 00064         BNEQ    7$                              ;
                    50              64          01  C3 00066         SUBL3   #1, BUF_I, R0                   ;
                                                02  18 0006A         BGEQ    6$                              ;
                                                50  D4 0006C         CLRL    R0                              ;
                                64              50  D0 0006E 6$:     MOVL    R0, BUF_I                       ;
                                                A3  11 00071         BRB     1$                              ;
                                15              51  91 00073 7$:     CMPB    CHAR, #21                       ; 0709
                                                9E  12 00076         BNEQ    1$                              ;
                                                64  D4 00078         CLRL    BUF_I                           ;
                                                9A  11 0007A         BRB     1$                              ; 0695
                                01              53  E8 0007C 8$:     BLBS    COMMAND_COMPLETE, 9$            ; 0718
                                                04  0007F         RET                                     ;
                                                52  94 00080 9$:     CLRB    IN_QUOTES                       ; 0724
                                50              01  CE 00082         MNEGL   #1, I                           ; 0725
                                                27  11 00085         BRB     13$                             ;
                                16              52  E8 00087 10$:    BLBS    IN_QUOTES, 11$                  ; 0726
                        61  8F          B0  A440    91 0008A         CMPB    COMMAND_BUF[I], #97             ; 0727
                                                0E  1F 00090         BLSSU   11$                             ;
                        7A  8F          B0  A440    91 00092         CMPB    COMMAND_BUF[I], #122            ;
                                                06  1A 00098         BGTRU   11$                             ;
                        B0  A440        20  8F      8A 0009A         BICB2   #-224, COMMAND_BUF[I]           ; 0728
                                                51  D4 000A0 11$:     CLRL    R1                              ; 0729
                                22          B0  A440    91 000A2         CMPB    COMMAND_BUF[I], #34             ;
                                                02  12 000A7         BNEQ    12$                             ;
                                                51  D6 000A9         INCL    R1                              ;
                                52              51  C8 000AB 12$:     XORB2   R1, IN_QUOTES                   ;
                    D5          50              64  F2 000AE 13$:     AOBLSS  BUF_I, I, 10$                   ; 0725
                                0C  A4          64  D0 000B2         MOVL    BUF_I, TPARSE_BLOCK+8           ; 0732
                                10  A4      B0  A4  9E 000B6         MOVAB   COMMAND_BUF, TPARSE_BLOCK+12    ; 0733
                                        28  A4  D4 000BB             CLRL    COMMAND_PROC                    ; 0734
                                    0000'   CF  9F 000BE             PUSHAB  COMMAND_KEY                     ; 0735
                                    0000'   CF  9F 000C2             PUSHAB  COMMAND_STATE                   ;
```

```
                                 04   A4  9F 000C6          PUSHAB  TPARSE_BLOCK                                    ;
                    00000000G    00   03  FB 000C9          CALLS   #3, LIB$TPARSE                                 ;
                                 52   50  D0 000D0          MOVL    R0, STATUS                                     ;
                    00000000G    8F   52  D1 000D3          CMPL    STATUS, #LIB$_SYNTAXERR                        ; 0740
                                      0D  12 000DA          BNEQ    14$                                            ;
                          00000000G  8F   DD 000DC          PUSHL   #PHN$_BADCMD                                   ; 0741
                              0000G  CF   01  FB 000E2      CALLS   #1, PHN$INFORM                                 ;
                                      18  11 000E7          BRB     16$                                            ;
                                 09   52  E8 000E9 14$:     BLBS    STATUS, 15$                                    ; 0743
                                      52  DD 000EC          PUSHL   STATUS                                         ;
                    00000000G    00   01  FB 000EE          CALLS   #1, LIB$SIGNAL                                 ;
                                 50       28  A4  D0 000F5 15$:     MOVL    COMMAND_PROC, R0                       ; 0744
                                      06  13 000F9          BEQL    16$                                            ;
                                      0C  A4  9F 000FB      PUSHAB  TPARSE_BLOCK+8                                 ; 0745
                                 60   01  FB 000FE          CALLS   #1, (R0)                                       ;
                                           0000' CF  9F 00101 16$:     PUSHAB  P.AAG                                  ; 0750
                                                 7E  D4 00105          CLRL    -(SP)                                  ;
                              0000G  CF   02  FB 00107      CALLS   #2, PHN$SHOW_TEXT                              ;
                                      64  D4 0010C          CLRL    BUF_I                                          ; 0751
                                          04 0010E          RET                                                    ; 0754

; Routire Size:  271 bytes,    Routine Base:  $CODE$ + 013F


;  435          0755  1
;  436          0756  0 end eludom


                                                                .EXTRN  LIB$SIGNAL

                                PSECT SUMMARY

     Name                       Bytes                          Attributes

$OWN$                            212  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
$PLIT$                            56  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
$CODE$                           590  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
_LIB$KEY0$                        24  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
_LIB$STATE$                      184  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
_LIB$KEY1$                        80  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)


                              Library Statistics

                                    --------  Symbols  --------      Pages       Processing
     File                           Total     Loaded   Percent      Mapped      Time

_$255$DUA28:[SYSLIB]STARLET.L32;1    9776          6         0         581        00:00.7
_$255$DUA28:[SYSLIB]TPAMAC.L32;1       42         24        57          14        00:00.1
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         590 code + 556 data bytes
; Run Time:        00:16.8
; Elapsed Time:    01:03.8
; Lines/CPU Min:    2703
; Lexemes/CPU-Min: 87343
; Memory Used:  200 pages
; Compilation Complete
```

PHONE
LIS

NETSLAVE
LIS

LINKSUBS
LIS

PHONEMSGS
LIS

STACKCMDS
LIS

TERMINAL
LIS

MISCCMDS
LIS

PUBSUBS
LIS

INPUT
LIS