```
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
```

FILEID**BASICCMDS

```
BBBBBBBB      AAAAAA      SSSSSSSS    IIIIII      CCCCCCCC    CCCCCCCC  MM      MM  DDDDDDDD      SSSSSSSS
BBBBBBBB      AAAAAA      SSSSSSSS    IIIIII      CCCCCCCC    CCCCCCCC  MM      MM  DDDDDDDD      SSSSSSSS
BB      BB   AA      AA  SS            II      CC          CC          MMMM  MMMM  DD      DD  SS
BB      BB   AA      AA  SS            II      CC          CC          MMMM  MMMM  DD      DD  SS
BB      BB   AA      AA  SS            II      CC          CC          MM  MM  MM  DD      DD  SS
BB      BB   AA      AA  SS            II      CC          CC          MM  MM  MM  DD      DD  SS
BBBBBBBB     AA      AA    SSSSSS      II      CC          CC          MM      MM  DD      DD    SSSSSS
BBBBBBBB     AA      AA    SSSSSS      II      CC          CC          MM      MM  DD      DD    SSSSSS
BB      BB   AAAAAAAAAA          SS    II      CC          CC          MM      MM  DD      DD          SS
BB      BB   AAAAAAAAAA          SS    II      CC          CC          MM      MM  DD      DD          SS
BB      BB   AA      AA          SS    II      CC          CC          MM      MM  DD      DD          SS
BB      BB   AA      AA          SS    II      CC          CC          MM      MM  DD      DD          SS
BBBBBBBB     AA      AA  SSSSSSSS    IIIIII      CCCCCCCC    CCCCCCCC  MM      MM  DDDDDDDD    SSSSSSSS
BBBBBBBB     AA      AA  SSSSSSSS    IIIIII      CCCCCCCC    CCCCCCCC  MM      MM  DDDDDDDD    SSSSSSSS

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 'BASICCMDS - Basic Phone Commands'
;    2          0002  0         module basiccmds (
;    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, Basic Phone Commands
;   32          0032  1 !
;   33          0033  1 ! Abstract:     This module handles the basic essential user commands:
;   34          0034  1 !                         DIAL    Call someone on the phone.
;   35          0035  1 !                         ANSWER  Answer the phone.
;   36          0036  1 !                         REJECT  Reject a call.
;   37          0037  1 !                         HANGUP  Hang up the phone.
;   38          0038  1 !
;   39          0039  1 !
;   40          0040  1 ! Environment:
;   41          0041  1 !
;   42          0042  1 ! Author: Paul C. Anagnostopoulos, Creation Date: 12 November 1980
;   43          0043  1 !
;   44          0044  1 ! Modified By:
;   45          0045  1 !
;   46          0046  1 !       V03-003 BLS0251         Benn Schreiber          8-Dec-1983
;   47          0047  1 !               Use $BRKTHRU rather than $BRDCST.
;   48          0048  1 !
;   49          0049  1 !       V03-002 MHB0084         Mark Bramhall           17-Jan-1983
;   50          0050  1 !               PHN$ESTAB_LINK now always displays its status via
;   51          0051  1 !               PHN$INFORM; its callers just check the returned status.
;   52          0052  1 !
;   53          0053  1 !       V03-001 PCA0040         Paul Anagnostopoulos    26-Mar-1982
;   54          0054  1 !               Major changes to convert from process name to user name.
;   55          0055  1 !--
```

```
;    57          0056  1 %sbttl 'Module Declarations'
;    58          0057  1 !
;    59          0058  1 ! Libraries and Requires:
;    60          0059  1 !
;    61          0060  1
;    62          0061  1 library 'sys$library:starlet.l32';
;    63          0062  1 require 'phonereq';
;    64          0391  1
;    65          0392  1 !
;    66          0393  1 ! Table of Contents:
;    67          0394  1 !
;    68          0395  1
;    69          0396  1 forward routine
;    70          0397  1         phn$dial_cmd: novalue,
;    71          0398  1         phn$ring_out: novalue,
;    72          0399  1         phn$local_jangle,
;    73          0400  1         phn$rang_in: novalue,
;    74          0401  1         phn$jangle_string,
;    75          0402  1         phn$answer_cmd: novalue,
;    76          0403  1         phn$answered: novalue,
;    77          0404  1         phn$reject_cmd: novalue,
;    78          0405  1         phn$rejected: novalue,
;    79          0406  1         phn$busy: novalue,
;    80          0407  1         phn$talk: novalue,
;    81          0408  1         phn$listen: novalue,
;    82          0409  1         phn$hangup_cmd: novalue,
;    83          0410  1         phn$hungup: novalue;
;    84          0411  1
;    85          0412  1
;    86          0413  1 !
;    87          0414  1 ! External References:
;    88          0415  1 !
;    89          0416  1
;    90          0417  1 external routine
;    91          0418  1         phn$break_call,
;    92          0419  1         phn$break_link,
;    93          0420  1         phn$cmp_target,
;    94          0421  1         phn$estab_link,
;    95          0422  1         phn$exit_cmd,
;    96          0423  1         phn$force_links,
;    97          0424  1         phn$fresh_screen,
;    98          0425  1         phn$inform,
;    99          0426  1         phn$make_tsb,
;   100          0427  1         phn$prepare_users_target,
;   101          0428  1         phn$queue_smb,
;   102          0429  1         phn$read_slave,
;   103          0430  1         phn$redundant,
;   104          0431  1         phn$send_smb,
;   105          0432  1         phn$show_text,
;   106          0433  1         phn$term_characteristic;
;   107          0434  1
;   108          0435  1 !
;   109          0436  1 ! Own Variables:
;   110          0437  1 !
```

```
;  112          0438  1 %sbttl 'PHN$DIAL_CMD - Call Someone on the Phone'
;  113          0439  1 !++
;  114          0440  1 ! Functional Description:
;  115          0441  1 !       This routines handles the DIAL command, used to call someone on the
;  116          0442  1 !       phone.
;  117          0443  1 !
;  118          0444  1 ! Formal Parameters:
;  119          0445  1 !       target          Address of descriptor of the argument to the DIAL
;  120          0446  1 !                       command as entered by the user.
;  121          0447  1 !
;  122          0448  1 ! Implicit Inputs:
;  123          0449  1 !       global data
;  124          0450  1 !
;  125          0451  1 ! Implicit Outputs:
;  126          0452  1 !       global data
;  127          0453  1 !
;  128          0454  1 ! Returned Value:
;  129          0455  1 !       none
;  130          0456  1 !
;  131          0457  1 ! Side Effects:
;  132          0458  1 !
;  133          0459  1 !--                                                                              ;
;  134          0460  1
;  135          0461  1
;  136          0462  2 global routine phn$dial_cmd(target): novalue = begin
;  137          0463  2
;  138          0464  2 local
;  139          0465  2         status: long,
;  140          0466  2         op: ref pub,                    ! Points to our PUB.
;  141          0467  2         tp: ref pub;                    ! Points to target's PUB.
;  142          0468  2 local
;  143          0469  2         local_described_buffer(prepared_target,nam$c_maxrss);
;  144          0470  2
;  145          0471  2
;  146          0472  2 ! We begin by making sure that our phone is not busy receiving a call
;  147          0473  2 ! from someone else.
;  148          0474  2
;  149          0475  2 op = .phn$gq_pubhead[0];
;  150          0476  3 if .op[pub_v_answering] then (
;  151          0477  3         phn$inform(phn$_phonebusy);
;  152          0478  3         return;
;  153          0479  2 );
;  154          0480  2
;  155          0481  2 ! Now we will prepare the target specified by the user.  This gets it
;  156          0482  2 ! ready for use in establishing a link to the person.
;  157          0483  2
;  158          0484  2 phn$prepare_users_target(.target,false,prepared_target);
;  159          0485  2
;  160          0486  2 ! We now attempt to establish a link to the target person.  If this fails,
;  161          0487  2 ! we will get a status back that we can use to display a message.
;  162          0488  2
;  163          0489  2 status = phn$estab_link(prepared_target,tp);
;  164          0490  2 if .status nequ phn$_ok then
;  165          0491  2         return;
;  166          0492  2
;  167          0493  2 ! Now we have to see if we are establishing a redundant link, that is,
;  168          0494  2 ! a link to someone we are already linked to.  If so, forget it.
```

```
;  169          0495  2
;  170          0496  3 if phn$redundant(.tp) then (
;  171          0497  3         phn$break_link(.tp,smb__busy);
;  172          0498  3         phn$inform(phn$_ivreduncall);
;  173          0499  3         return;
;  174          0500  2 );
;  175          0501  2
;  176          0502  2 ! Finally, we set up our PUB so we'll know who we're calling. We also queue
;  177          0503  2 ! a steering message to do a first-time ring of the person's phone.
;  178          0504  2
;  179          0505  2 op[pub_v_calling] = true;
;  180          0506  2 op[pub_l_busylink] = .tp;
;  181          0507  2 phn$queue_smb(smb__ring_out,describe(%char(true)));
;  182          0508  2
;  183          0509  2 return;
;  184          0510  2
;  185          0511  1 end;


                                                .TITLE  BASICCMDS BASICCMDS - Basic Phone Commands
                                                .IDENT  \V04-000\

                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                      01  00000 P.AAB:  .ASCII  <1>                                          ;
                                          00001        .BLKB   3
                                00000001  00004 P.AAA:  .LONG   1                                           ;
                                00000000' 00008        .ADDRESS P.AAB                                       ;

                                                .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$GB_FLAGS, PHN$BREAK_CALL
                                                        .EXTRN  PHN$BREAK_LINK, PHN$CMP_TARGET
                                                        .EXTRN  PHN$ESTAB_LINK, PHN$EXIT_CMD
                                                        .EXTRN  PHN$FORCE_LINK$
                                                        .EXTRN  PHN$FRESH_SCREEN
                                                        .EXTRN  PHN$INFORM, PHN$MAKE_TSB
                                                        .EXTRN  PHN$PREPARE_USERS_TARGET
                                                        .EXTRN  PHN$QUEUE_SMB, PHN$READ_SLAVE
                                                        .EXTRN  PHN$REDUNDANT, PHN$SEND_SMB
                                                        .EXTRN  PHN$SHOW_TEXT, PHN$TERM_CHARACTERISTIC

                                                        .PSECT  $CODE$,NOWRT,2

                                                 0004 00000     .ENTRY  PHN$DIAL_CMD, Save R2                           ; 0462
                                             5E      FEF4    CE  9E 00002    MOVAB   -268(SP), SP
                                          04 AE        FF    8F  9A 00007    MOVZBL  #255, PREPARED_TARGET           ; 0469
                                          08 AE        0C    AE  9E 0000C    MOVAB   PREPARED_TARGET+8, PREPARED_TARGET+4
                                             52      0000G   CF  D0 00011    MOVL    PHN$GQ_PUBHEAD, OP              ; 0475
                                  08     00F0    C2          04  E1 00016    BBC     #4, 240(OP), 1$                 ; 0476
                                              00000000G   8F  DD 0001C    PUSHL   #PHN$_PHONEBUSY                 ; 0477
                                                          3A  11 00022    BRB     2$
                                                 04     AE  9F 00024 1$:    PUSHAB  PREPARED_TARGET                 ; 0484
                                                        7E  D4 00027    CLRL    -(SP)
                                                 04     AC  DD 00029    PUSHL   TARGET
                                           0000G CF             03  FB 0002C    CALLS   #3, PHN$PREPARE_USERS_TARGET
                                                        5E  DD 00031    PUSHL   SP                              ; 0489
                                                 08     AE  9F 00033    PUSHAB  PREPARED_TARGET
                                           0000G CF             02  FB 00036    CALLS   #2, PHN$ESTAB_LINK
                                      00000000G  8F             50  D1 0003B    CMPL    STATUS, #PHN$_OK                ; 0490
                                                          35  12 00042    BNEQ    4$                              ; 0496
                                                          6E  DD 00044    PUSHL   TP
                                           0000G CF             01  FB 00046    CALLS   #1, PHN$REDUNDANT
                                              16                50  E9 0004B    BLBC    R0, 3$                          ; 0497
                                                          0A  DD 0004E    PUSHL   #10
                                                    04    AE  DD 00050    PUSHL   TP
                                           0000G CF             02  FB 00053    CALLS   #2, PHN$BREAK_LINK
                                                 00000000G  8F  DD 00058    PUSHL   #PHN$_IVREDUNCALL               ; 0498
                                           0000G CF             01  FB 0005E 2$:    CALLS   #1, PHN$INFORM
                                                          04  00063    RET                                     ; 0496
                                           00F0  C2             08  88 00064 3$:    BISB2   #8, 240(OP)             ; 0505
                                           00F8  C2                 6E  D0 00069    MOVL    TP, 248(OP)             ; 0506
                                                    0000'  CF   9F  0006E    PUSHAB  P.AAA                           ; 0507
                                                          06  DD 00072    PUSHL   #6
                                           0000G CF             02  FB 00074    CALLS   #2, PHN$QUEUE_SMB
                                                          04  00079 4$:    RET                                     ; 0511

; Routine Size:  122 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  187          0512  1 %sbttl 'PHN$RING_OUT - Ring Someones Phone'
;  188          0513  1 !++
;  189          0514  1 ! Functional Description:
;  190          0515  1 !       This steering message routine is called to ring someone's
;  191          0516  1 !       telephone.  We have to distinguish between first-time rings
;  192          0517  1 !       and subsequent rings, as described below.
;  193          0518  1 !
;  194          0519  1 ! Formal Parameters:
;  195          0520  1 !       first_ring              The address of a descriptor of a byte flag:
;  196          0521  1 !                                       true    First-time ring.
;  197          0522  1 !                                       false   Subsequent ring.
;  198          0523  1 !
;  199          0524  1 ! Implicit Inputs:
;  200          0525  1 !       global data
;  201          0526  1 !
;  202          0527  1 ! Implicit Outputs:
;  203          0528  1 !       global data
;  204          0529  1 !
;  205          0530  1 ! Returned Value:
;  206          0531  1 !       none
;  207          0532  1 !
;  208          0533  1 ! Side Effects:
;  209          0534  1 !
;  210          0535  1 !--
;  211          0536  1
;  212          0537  1
;  213          0538  2 global routine phn$ring_out(first_ring): novalue = begin
;  214          0539  2
;  215          0540  2 bind
;  216          0541  2         first_ring_dsc  = .first_ring: descriptor,
;  217          0542  2         first_ring_flag = .first_ring_dsc[ptr]: byte;
;  218          0543  2
;  219          0544  2 local
;  220          0545  2         status: long,
;  221          0546  2         op: ref pub,                    ! Points to our PUB.
;  222          0547  2         tp: ref pub;                    ! Points to target's PUB.
;  223          0548  2
;  224          0549  2
;  225          0550  2 ! This little routine is the timer AST.  A timer goes off when we need
;  226          0551  2 ! to do subsequent rings of a person's phone.  Just queue another ring_out
;  227          0552  2 ! steering message with a flag that says "subsequent ring".
;  228          0553  2
;  229          0554  3 routine timer_ast: novalue = begin
;  230          0555  3
;  231          0556  3 phn$queue_smb(smb__ring_out,describe(%char(false)));
;  232          0557  3 return;
;  233          0558  3
;  234          0559  2 end;


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                      00  0000C P.AAD:   .ASCII  <0>                                  ;
                                          0000D         .BLKB   3
                                00000001  00010 P.AAC:   .LONG   1                                    ;
                                00000000' 00014         .ADDRESS P.AAD                                ;
```

```
                                                        .PSECT  $CODE$,NOWRT,2

                                              0000 00000 TIMER_AST:
                                                        .WORD   Save nothing                    ; 0554
                                        0000'  CF  9F 00002     PUSHAB  P.AAC                   ; 0556
                                               06  DD 00006     PUSHL   #6                      ;
                                    0000G  CF          02  FB 00008     CALLS   #2, PHN$QUEUE_SMB  ;
                                                       04 0000D     RET                        ; 0559

; Routine Size:  14 bytes,    Routine Base:  $CODE$ + 007A
```

```
;  236     0560  2 ! First we have to ensure that we really are still trying to call someone.
;  237     0561  2
;  238     0562  2 op = .phn$gg_pubhead[0];
;  239     0563  2 if (not .op[pub_v_calling]) or (.op[pub_l_busylink] eqla 0) then
;  240     0564  2         return;
;  241     0565  2
;  242     0566  2 ! Now we split up depending on whether this is a local or remote ring.
;  243     0567  2
;  244     0568  2 tp = .op[pub_l_busylink];
;  245     0569  3 begin
;  246     0570  3         bind target_tsb = tp[pub_b_tsb]: tsb;
;  247     0571  3
;  248     0572  4 if not .target_tsb[tsb_v_remote] then (
;  249     0573  4
;  250     0574  4         ! We are ringing a local phone.  Call a routine to do it, and then
;  251     0575  4         ! check the status.  If bad, we break the call and tell the user why.
;  252     0576  4
;  253     0577  4         status = phn$local_jangle(.tp,.first_ring_flag);
;  254     0578  5         if .status nequ phn$_ok then (
;  255     0579  5                 phn$break_call();
;  256     0580  5                 phn$inform(.status);
;  257     0581  5                 return;
;  258     0582  4         );
;  259     0583  4
;  260     0584  4 ) else (
;  261     0585  4
;  262     0586  4         ! It appears we are ringing a remote node, which we obviously can't
;  263     0587  4         ! do ourselves.  Send a steering message to the remote slave, who
;  264     0588  4         ! will try to ring the person.  It will send us back a status.
;  265     0589  4
;  266     0590  4         local
;  267     0591  4                 local_described_buffer(ring_status,4);
;  268     0592  4
;  269     0593  4         phn$send_smb(.tp,smb__rang_in,first_ring_dsc);
;  270     0594  4         status = phn$read_slave(.tp[pub_w_channel],ring_status,true);
;  271     0595  5         if .status nequ phn$_ok then (
;  272     0596  5                 phn$break_call();
;  273     0597  5                 phn$inform(.status);
;  274     0598  5                 return;
;  275     0599  4         );
;  276     0600  5         if ..ring_status[ptr] nequ phn$_ok then (
;  277     0601  5                 phn$break_call();
;  278     0602  5                 phn$inform(..ring_status[ptr]);
;  279     0603  5                 return;
;  280     0604  4         );
;  281     0605  3 );
;  282     0606  3
;  283     0607  3 ! Finally, we inform the user that we rang the person's phone, and we set
;  284     0608  3 ! a timer for 10 seconds.  When the timer goes off, we'll ring the guy again.
;  285     0609  3
;  286     0610  3 phn$inform(phn$_justrang,target_tsb[tsb_q_tkndsc,
;  287     0611  3                                     .target_tsb[tsb_w_tkncount]]);
;  288   P 0612  3 status = $setimr(daytim=uplit long(-10*10000000,-1),
;  289     0613  3                   astadr=timer_ast);
;  290     0614  3 check (.status);
;  291     0615  3 return;
;  292     0616  2 end;
```

```
;  293          0617  2
;  294          0618  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                FFFFFFFF  FA0A1F00  00018 P.AAE:.LONG   -100000000, -1                                  ;
                                                                .EXTRN  SYS$SETIMR

                                                                .PSECT  $CODE$,NOWRT,2

                                                    007C 00000  .ENTRY  PHN$RING_OUT, Save R2,R3,R4,R5,R6               ; 0538
                                    56 00000000G    8F   D0 00002       MOVL    #PHN$_OK, R6                            ;
                                    5E              0C   C2 00009       SUBL2   #12, SP                                 ;
                                    53        04    AC   D0 0000C       MOVL    FIRST_RING, R3                          ; 0541
                                    50      0000G   CF   D0 00010       MOVL    PHN$GQ_PUBHEAD, OP                      ; 0562
                          01    00F0    C0         03   E0 00015       BBS     #3, 240(OP), 1$                         ; 0563
                                                    04   0001B       RET                                             ;
                                        00F8    C0   D5 0001C 1$:    TSTL    248(OP)                                 ;
                                                01   12 00020       BNEQ    2$                                      ;
                                                    04   00022       RET                                             ;
                                    52  00F8    C0   D0 00023 2$:    MOVL    248(OP), TP                             ; 0568
                                    54    0C    A2   9E 00028       MOVAB   12(TP), R4                              ; 0570
                                    15          64   E8 0002C       BLBS    (R4), 3$                                ; 0572
                                    7E    04    B3   9A 0002F       MOVZBL  @4(R3), -(SP)                           ; 0577
                                                52   DD 00033       PUSHL   TP                                      ;
                              0000V CF          02   FB 00035       CALLS   #2, PHN$LOCAL_JANGLE                    ;
                                    55          50   D0 0003A       MOVL    R0, STATUS                              ;
                                    56          55   D1 0003D       CMP     STATUS, R6                              ; 0578
                                                49   13 00040       BEQ     7$                                      ;
                                                2A   11 00042       BRB     4$                                      ; 0579
                                    6E          04   D0 00044 3$:    MOVL    #4, RING_STATUS                         ; 0591
                              04    AE    08    AE   9E 00047       MOVAB   RING_STATUS+8, RING_STATUS+4             ;
                                                53   DD 0004C       PUSHL   R3                                      ; 0593
                                                08   DD 0004E       PUSHL   #8                                      ;
                                                52   DD 00050       PUSHL   TP                                      ;
                            0000G CF            03   FB 00052       CALLS   #3, PHN$SEND_SMB                        ;
                                                01   DD 00057       PUSHL   #1                                      ; 0594
                                          04    AE   9F 00059       PUSHAB  RING_STATUS                             ;
                                    7E    00F4  C2   3C 0005C       MOVZWL  244(TP), -(SP)                          ;
                            0000G   CF          03   FB 00061       CALLS   #3, PHN$READ_SLAVE                      ;
                                    55          50   D0 00066       MOVL    R0, STATUS                              ;
                                    56          55   D1 00069       CMPL    STATUS, R6                              ; 0595
                                                09   13 0006C       BEQL    5$                                      ;
                            0000G   CF          00   FB 0006E 4$:    CALLS   #0, PHN$BREAK_CALL                      ; 0596
                                                55   DD 00073       PUSHL   STATUS                                  ; 0597
                                                0E   11 00075       BRB     6$                                      ;
                                    56    04    BE   D1 00077 5$:    CMPL    @RING_STATUS+4, R6                      ; 0600
                                                0E   13 0007B       BEQL    7$                                      ;
                            0000G   CF          00   FB 0007D       CALLS   #0, PHN$BREAK_CALL                      ; 0601
                                          04    BE   DD 00082       PUSHL   @RING_STATUS+4                          ; 0602
                            0000G   CF          01   FB 00085 6$:    CALLS   #1, PHN$INFORM                          ;
                                                    04   0008A       RET                                             ; 0600
                                    50    02    A4   3C 0008B 7$:    MOVZWL  2(R4), R0                               ; 0611
                                          04  A440  7F   0008F       PUSHAQ  4(R4)[R0]                               ;
```

```
                                00000000G  8F  DD 00093          PUSHL   #PHN$_JUSTRANG                    ;
                          0000G CF         02  FB 00099          CALLS   #2, PHN$INFORM                   ;
                                           7E  D4 0009E          CLRL    -(SP)                            ; 0613
                                    FF4E   CF  9F 000A0          PUSHAB  TIMER_AST                        ;
                                    0000'  CF  9F 000A4          PUSHAB  P.AAE                            ;
                                           7E  D4 000A8          CLRL    -(SP)                            ;
                    00000000G  00          04  FB 000AA          CALLS   #4, SYS$SETIMR                   ;
                               55          50  DC 000B1          MOVL    R0, STATUS                       ;
                               09          55  E8 000B4          BLBS    STATUS, 8$                       ; 0614
                                           55  DD 000B7          PUSHL   STATUS                           ;
                    00000000G  00          01  FB 000B9          CALLS   #1, LIB$SIGNAL                   ;
                                           04  000C0 8$:         RET                                      ; 0618

; Routine Size:  193 bytes,    Routine Base:  $CODE$ + 0088
```

```
;  296          0619  1 %sbttl 'PHN$LOCAL_JANGLE - Produce Actual Phone Ring'
;  297          0620  1 !++
;  298          0621  1 ! Functional Description:
;  299          0622  1 !       This routine is called to actually ring the phone of a local person.
;  300          0623  1 !       This includes both broadcasting a ring and sending a ring message to
;  301          0624  1 !       the person's mailbox.
;  302          0625  1 !
;  303          0626  1 ! Formal Parameters:
;  304          0627  1 !       target_pub      Address of the PUB describing the person to be rung.
;  305          0628  1 !       first_ring_flag A flag to say if this is the first ring:
;  306          0629  1 !                                       true    First-time ring.
;  307          0630  1 !                                       false   Subsequent ring.
;  308          0631  1 !
;  309          0632  1 ! Implicit Inputs:
;  310          0633  1 !       global data
;  311          0634  1 !
;  312          0635  1 ! Implicit Outputs:
;  313          0636  1 !       global data
;  314          0637  1 !
;  315          0638  1 ! Returned Value:
;  316          0639  1 !       phn$_loggedoff  The person has logged off the system.
;  317          0640  1 !       phn$_cantreach  The person can't be reached for some reason.
;  318          0641  1 !       phn$_targterm   None of the person's terminals are usable by PHONE.
;  319          0642  1 !       phn$_unplugged  The person has /NOBROADCAST set on all terminals
;  320          0643  1 !
;  321          0644  1 ! Side Effects:
;  322          0645  1 !
;  323          0646  1 !--
;  324          0647  1
;  325          0648  1
;  326          0649  2 global routine phn$local_jangle(target_pub,first_ring_flag) = begin
;  327          0650  2
;  328          0651  2 bind
;  329          0652  2         tp = .target_pub: pub,
;  330          0653  2         target_tsb = tp[pub_b_tsb]: tsb,
;  331          0654  2         target_name = target_tsb[tsb_q_tkndsc,.target_tsb[tsb_w_tkncount]]: descriptor;
;  332          0655  2
;  333          0656  2 own
;  334          0657  2         process_pid: long,
;  335          0658  2         own_described_buffer(user_name,12),
;  336          0659  2         parent_pid: long,
;  337          0660  2         own_described_buffer(term_number,7),
;  338          0661  2         own_described_buffer(image_name,nam$c_maxrss);
;  339          0662  2 bind
;  340          0663  2         get_proc = uplit(word(4),word(jpi$_pid),
;  341          0664  2                          long(process_pid),
;  342          0665  2                          long(0),
;  343          0666  2                          word(12),word(jpi$_username),
;  344          0667  2                          long(user_name+8),
;  345          0668  2                          long(user_name),
;  346          0669  2                          word(4),word(jpi$_owner),
;  347          0670  2                          long(parent_pid),
;  348          0671  2                          long(0),
;  349          0672  2                          word(7),word(jpi$_terminal),
;  350          0673  2                          long(term_number+8),
;  351          0674  2                          long(term_number),
;  352          0675  2                          long(0)),
```

```
;  353          0676  2          get_image = uplit(word(nam$c_maxrss),word(jpi$_imagname),
;  354          0677  2                            long(image_name+8),
;  355          0678  2                            long(image_name),
;  356          0679  2                            long(0));
;  357          0680  2
;  358          0681  2 local
;  359          0682  2          status: long,
;  360          0683  2          wild_pid: long,
;  361          0684  2          potential: long, usable: long, jangled: long, active: long;
;  362          0685  2
;  363          0686  2
;  364          0687  2 ! We are doing a single local phone ring.  This requires us to find all
;  365          0688  2 ! the processes for the target user and ring each one that we can.
;  366          0689  2
;  367          0690  2 potential = usable = jangled = active = 0;
;  368          0691  2 wild_pid = -1;
;  369          0692  3 loop (
;  370          0693  3
;  371          0694  3          ! Get the information on the next process.  If there aren't any
;  372          0695  3          ! more, then we are done.
;  373          0696  3
;  374        P 0697  3          status = $getjpi(efn=phn$k_getjpiefn,
;  375        P 0698  3                           pidadr=wild_pid,
;  376          0699  3                           itmlst=get_proc);
;  377          0700  3
;  378          0701  3 exitif (.status eqlu ss$_nomoreproc);
;  379          0702  3
;  380          0703  3          ! If we got a process, then determine if it is a detached
;  381          0704  3          ! interactive process owned by the target.
;  382          0705  3
;  383          0706  4          if .status eqlu ss$_normal then (
;  384          0707  4                  status = $waitfr(efn=phn$k_getjpiefn);
;  385          0708  4                  check (.status);
;  386          0709  4
;  387          0710  4                  if ch$eql(.target_name[len],.target_name[ptr], .user_name[len],.user_name[ptr],' ') and
;  388          0711  4                     .parent_pid eqlu 0 and
;  389          0712  5                     .term_number[len] nequ 0 then (
;  390          0713  5
;  391          0714  5                          ! We got a potential candidate.  Make sure that
;  392          0715  5                          ! their terminal is usable by PHONE.
;  393          0716  5
;  394          0717  5                          inc (potential);
;  395          0718  6                          if phn$term_characteristic(term_number,tt$m_scope) then (
;  396          0719  6                                  inc (usable);
;  397          0720  6
;  398          0721  6                                  ! An interesting process.  If they aren't
;  399          0722  6                                  ! running PHONE, try to broadcast a ring
;  400          0723  6                                  ! message to them.  If they are, just remember
;  401          0724  6                                  ! that fact.
;  402          0725  6
;  403        P 0726  6                                  status = $getjpi(efn=phn$k_getjpiefn,
;  404        P 0727  6                                                   pidadr=process_pid,
;  405          0728  6                                                   itmlst=get_image);
;  406          0729  7                                  if .status eqlu ss$_normal then (
;  407          0730  7                                          status = $waitfr(efn=phn$k_getjpiefn);
;  408          0731  7                                          check (.status);
;  409          0732  7                                          if ch$find_sub(.image_name[len],.image_name[ptr],
```

```
;  410          0733  8                                                   9,uplit byte('PHONE.EXE'))       eqla 0 then (
;  411          0734  8                                             if phn$jangle_string(.phn$gq_pubhead[0],term_number) then
;  412          0735  8                                                     inc (jangled);
;  413          0736  7                                             ) else
;  414          0737  7                                                     inc (active);
;  415          0738  6                                         );
;  416          0739  5                                 );
;  417          0740  4                         );
;  418          0741  3                 );
;  419          0742  2 );
;  420          0743  2
;  421          0744  2 ! Now we just return if there was some trouble.  Various statuses are
;  422          0745  2 ! returned depending on the exact problem.
;  423          0746  2
;  424          0747  2 if .potential eqlu 0 then
;  425          0748  2         return phn$_loggedoff;
;  426          0749  2 if .usable eqlu 0 then
;  427          0750  2         return phn$_targterm;
;  428          0751  2 if .jangled + .active eqlu 0 then
;  429          0752  2         return phn$_unplugged;
;  430          0753  2
;  431          0754  2 ! OK.  Now we may need to send a rang_in message to the target's
;  432          0755  2 ! receive mailbox.  This is always done on the first ring, so that
;  433          0756  2 ! no matter what they're doing, PHONE will get our ring.  Also, on
;  434          0757  2 ! each subsequent ring, we want to send a message if they are now
;  435          0758  2 ! running PHONE, so they'll get another buzz.
;  436          0759  2
;  437          0760  3 if .first_ring_flag or .active nequ 0 then (
;  438          0761  3
;  439          0762  3         local
;  440          0763  3                 first_ring_dsc: descriptor;
;  441          0764  3
;  442          0765  3         first_ring_dsc[len] = 1;
;  443          0766  3         first_ring_dsc[ptr] = first_ring_flag;
;  444          0767  3         phn$send_smb(tp,smb__rang_in,first_ring_dsc),
;  445          0768  2 );
;  446          0769  2
;  447          0770  2 return phn$_ok;
;  448          0771  2
;  449          0772  1 end;


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    0004  00020 P.AAF:  .WORD   4                                                    ;
                                    0319  00022         .WORD   793                                                  ;
                                00000000' 00024         .ADDRESS PROCESS_PID                                         ;
                                00000000  00028         .LONG   0                                                    ;
                                    000C  0002C         .WORD   12                                                   ;
                                    0202  0002E         .WORD   514                                                  ;
                                00000000' 00030         .ADDRESS USER_NAME+8                                         ;
                                00000000' 00034         .ADDRESS USER_NAME                                           ;
                                    0004  00038         .WORD   4                                                    ;
                                    0303  0003A         .WORD   771                                                  ;
                                00000000' 0003C         .ADDRESS PARENT_PID                                          ;
                                00000000  00040         .LONG   0                                                    ;
```

```
                                        0007  00044          .WORD   7
                                        031D  00046          .WORD   797
                                    00000000' 00048          .ADDRESS TERM_NUMBER+8
                                    00000000' 0004C          .ADDRESS TERM_NUMBER
                                    00000000  00050          .LONG   0
                                        00FF  00054 P.AAG:   .WORD   255
                                        0207  00056          .WORD   519
                                    00000000' 00058          .ADDRESS IMAGE_NAME+8
                                    00000000' 0005C          .ADDRESS IMAGE_NAME
                                    00000000  00060          .LONG   0
                45  58  45  2E  45  4E  4F  48  50  00064 P.AAH:   .ASCII  \PHONE.EXE\

                                                             .PSECT  $OWN$,NOEXE,2

                                              00000 PROCESS_PID:
                                                             .BLKB   4
                                        0000000C  00004 USER_NAME:
                                                             .LONG   12
                                    00000000' 00008          .ADDRESS USER_NAME+8
                                              0000C          .BLKB   12
                                              00018 PARENT_PID:
                                                             .BLKB   4
                                        00000007  0001C TERM_NUMBER:
                                                             .LONG   7
                                    00000000' 00020          .ADDRESS TERM_NUMBER+8
                                              00024          .BLKB   7
                                              0002B          .BLKB   1
                                        000000FF  0002C IMAGE_NAME:
                                                             .LONG   255
                                    00000000' 00030          .ADDRESS IMAGE_NAME+8
                                              00034          .BLKB   255

                                                       GET_PROC=         P.AAF
                                                       GET_IMAGE=        P.AAG
                                                             .EXTRN  SYS$GETJPI, SYS$WAITFR

                                                             .PSECT  $CODE$,NOWRT,2

                                            0FFC 00000       .ENTRY  PHN$LOCAL_JANGLE, Save R2,R3,R4,R5,R6,R7,-      ; 0649
                                                             R8,R9,R10,R11
                    5B 00000000G   00   9E 00002             MOVAB   SYS$GETJPI, R11
                    5A      0000'  CF   9E 00009             MOVAB   TERM_NUMBER, R10
                    5E             08   C2 0000E             SUBL2   #8, SP
            51      04     AC      0C   C1 00011             ADDL3   #12, TARGET_PUB, R1                             ; 0653
                    50     02      A1   3C 00016             MOVZWL  2(R1), R0                                       ; 0654
                    55     04  A140   7E 0001A             MOVAQ   4(R1)[R0], R5
                                   54   D4 0001F             CLRL    JANGLED                                         ; 0690
                                   57   7C 00021             CLRQ    ACTIVE
                                   59   D4 00023             CLRL    POTENTIAL
                    7E             01   CE 00025             MNEGL   #1, WILD_PID                                    ; 0691
                                   7E   7C 00028 1$:         CLRQ    -(SP)                                           ; 0699
                                   7E   D4 0002A             CLRL    -(SP)
                          0000'    CF   9F 0002C             PUSHAB  GET_PROC
                                   7E   D4 00030             CLRL    -(SP)
                            14     AE   9F 00032             PUSHAB  WILD_PID
                                   01   DD 00035             PUSHL   #1
                    6B             07   FB 00037             CALLS   #7, SYS$GETJPI
```

```
                                  56          50  D0 0003A          MOVL    R0, STATUS                                  ;
                        000009A8  8F          56  D1 0003D          CMPL    STATUS, #2472                               ; 0701
                                              03  12 00044          BNEQ    2$                                          ;
                                            00A0  31 00046          BRW     9$                                          ;
                                  01          56  D1 00049 2$:      CMPL    STATUS, #1                                  ; 0706
                                              DA  12 0004C          BNEQ    1$                                          ;
                                              01  DD 0004E          PUSHL   #1                                          ; 0707
                       00000000G  00          01  FB 00050          CALLS   #1, SYS$WAITFR                              ;
                                  56          50  D0 00057          MOVL    R0, STATUS                                  ;
                                  09          56  E8 0005A          BLBS    STATUS, 3$                                  ; 0708
                                              56  DD 0005D          PUSHL   STATUS                                      ;
                       00000000G  00          01  FB 0005F          CALLS   #1, LIB$SIGNAL                              ;
     E8    AA         20      04  B5          65  2D 00066 3$:      CMPC5   (R5), @4(R5), #32, USER_NAME, @USER_NAME+4  ; 0710
                                        EC    BA     0006D                                                              ;
                                              B7  12 0006F          BNEQ    1$                                          ;
                                        FC    AA  D5 00071          TSTL    PARENT_PID                                  ; 0711
                                              B2  12 00074          BNEQ    1$                                          ;
                                              6A  B5 00076          TSTW    TERM_NUMBER                                 ; 0712
                                              AE  13 00078          BEQL    1$                                          ;
                                              59  D6 0007A          INCL    POTENTIAL                                   ; 0717
                                  7E    1000  8F  3C 0007C          MOVZWL  #4096, -(SP)                                ; 0718
                                              5A  DD 00081          PUSHL   R10                                         ;
                           0000G  CF          02  FB 00083          CALLS   #2, PHN$TERM_CHARACTERISTIC                 ;
                                  9D          50  E9 00088 4$:      BLBC    R0, 1$                                      ;
                                              58  D6 0008B          INCL    USABLE                                      ; 0719
                                              7E  7C 0008D          CLRQ    -(SP)                                       ; 0728
                                              7E  D4 0008F          CLRL    -(SP)                                       ;
                                       0000'  CF  9F 00091          PUSHAB  GET_IMAGE                                   ;
                                              7E  D4 00095          CLRL    -(SP)                                       ;
                                        E4    AA  9F 00097          PUSHAB  PROCESS_PID                                 ;
                                              01  DD 0009A          PUSHL   #1                                          ;
                                  6B          07  FB 0009C          CALLS   #7, SYS$GETJPI                              ;
                                  56          50  D0 0009F          MOVL    R0, STATUS                                  ;
                                  01          56  D1 000A2          CMPL    STATUS, #1                                  ; 0729
                                              81  12 000A5          BNEQ    1$                                          ;
                                              01  DD 000A7          PUSHL   #1                                          ; 0730
                       00000000G  00          01  FB 000A9          CALLS   #1, SYS$WAITFR                              ;
                                  56          50  D0 000B0          MOVL    R0, STATUS                                  ;
                                  09          56  E8 000B3          BLBS    STATUS, 5$                                  ; 0731
                                              56  DD 000B6          PUSHL   STATUS                                      ;
                       00000000G  00          01  FB 000B8          CALLS   #1, LIB$SIGNAL                              ;
     14    BA         10  AA   0000'  CF          09  39 000BF 5$:      MATCHC  #9, P.AAH, IMAGE_NAME, @IMAGE_NAME+4        ; 0733
                                              03  13 000C8          BEQL    6$                                          ;
                                  53          09  D0 000CA          MOVL    #9, R3                                      ;
                                  53          09  C2 000CD 6$:      SUBL2   #9, R3                                      ;
                                              12  12 000D0          BNEQ    7$                                          ;
                                              5A  DD 000D2          PUSHL   R10                                         ; 0734
                                       0000G  CF  DD 000D4          PUSHL   PHN$GQ_PUBHEAD                              ;
                           0000V  CF          02  FB 000D8          CALLS   #2, PHN$JANGLE_STRING                       ;
                                  A8          50  E9 000DD          BLBC    R0, 4$                                      ;
                                              54  D6 000E0          INCL    JANGLED                                     ; 0735
                                              02  11 000E2          BRB     8$                                          ; 0732
                                              57  D6 000E4 7$:      INCL    ACTIVE                                      ; 0737
                                            FF3F  31 000E6 8$:      BRW     1$                                          ; 0729
                                              59  D5 000E9 9$:      TSTL    POTENTIAL                                   ; 0747
                                              08  12 000EB          BNEQ    10$                                         ;
                            50 00000000G  8F  D0 000ED          MOVL    #PHN$_LOGGEDOFF, R0                         ; 0748
```

```
                                                04 000F4          RET
                                             58 D5 000F5 10$:     TSTL    USABLE                                  ; 0749
                                             08 12 000F7          BNEQ    11$
                                   50 00000000G 8F D0 000F9       MOVL    #PHN$_TARGTERM, R0                      ; 0750
                                                04 00100          RET
                                   54        57 C0 00101 11$:     ADDL2   ACTIVE, R4                              ; 0751
                                             08 12 00104          BNEQ    12$
                                   50 00000000G 8F D0 00106       MOVL    #PHN$_UNPLUGGED, R0                     ; 0752
                                                04 0010D          RET
                                   04     08    AC E8 0010E 12$:  BLBS    FIRST_RING_FLAG, 13$                    ; 0760
                                             57 D5 00112          TSTL    ACTIVE
                                             16 13 00114          BEQL    14$
                            04     AE           01 B0 00116 13$:  MOVW    #1, FIRST_RING_DSC                      ; 0765
                            08     AE     08    AC 9E 0011A       MOVAB   FIRST_RING_FLAG, FIRST_RING_DSC+4       ; 0766
                                          04    AE 9F 0011F       PUSHAB  FIRST_RING_DSC                          ; 0767
                                             08 DD 00122          PUSHL   #8
                                          04    AC DD 00124       PUSHL   TARGET_PUB
                          0000G    CF           03 FB 00127       CALLS   #3, PHN$SEND_SMB
                                   50 00000000G 8F D0 0012C 14$:  MOVL    #PHN$_OK, R0                            ; 0770
                                                04 00133          RET                                             ; 0772

; Routine Size:  308 bytes,    Routine Base:  $CODE$ + 0149
```

```
;  451          0773  1 %sbttl 'PHN$RANG_IN - Handle a Ring on Our Phone'
;  452          0774  1 !++
;  453          0775  1 ! Functional Description:
;  454          0776  1 !       This steering message routine handles the rang_in steering message,
;  455          0777  1 !       which someone sends us when they want to ring our phone.  We have
;  456          0778  1 !       to make sure our phone is not busy, and then we can display a
;  457          0779  1 !       nice message to the user.
;  458          0780  1 !
;  459          0781  1 ! Formal Pa ameters:
;  460          0782  1 !       message         The complete node/user name spec of the person
;  461          0783  1 !                       calling us.  Ends with an eofrom character.
;  462          0784  1 !                       This is followed by a flag which is set if this
;  463          0785  1 !                       is the first ring, clear otherwise.
;  464          0786  1 !
;  465          0787  1 ! Implicit Inputs:
;  466          0788  1 !       global data
;  467          0789  1 !
;  468          0790  1 ! Implicit Outputs:
;  469          0791  1 !       global data
;  470          0792  1 !
;  471          0793  1 ! Returned Value:
;  472          0794  1 !       none
;  473          0795  1 !
;  474          0796  1 ! Side Effects:
;  475          0797  1 !
;  476          0798  1 !--
;  477          0799  1
;  478          0800  1
;  479          0801  2 global routine phn$rang_in(message): novalue = begin
;  480          0802  2
;  481          0803  2 bind
;  482          0804  2         message_dsc = .message: descriptor;
;  483          0805  2
;  484          0806  2 local
;  485          0807  2         status: long,
;  486          0808  2         first_ring_flag: byte,
;  487          0809  2         caller_tsb: tsb,                ! TSB to parse caller's spec.
;  488          0810  2         op: ref pub,                    ! Pointer to our PUB.
;  489          0811  2         tp: ref pub;                    ! Pointer to caller's PUB.
;  490          0812  2
;  491          0813  2
;  492          0814  2 ! We begin by isolating the caller's spec and making a TSB to parse it.
;  493          0815  2 ! We also isolate the first-ring flag.
;  494          0816  2
;  495          0817  2         first_ring_flag = ch$rchar(.message_dsc[ptr]+.message_dsc[len]-1);
;  496          0818  2         message_dsc[len] = .message_dsc[len] - 2;
;  497          0819  2         status = phn$make_tsb(message_dsc,caller_tsb);
;  498          0820  2         check (.status);
;  499          0821  2
;  500          0822  2 ! Now we see if we are calling ourselves.  In this case we just answer the
;  501          0823  2 ! phone immediately.  There are two reasons for this action.  First, we can't
;  502          0824  2 ! require the user to enter an ANSWER command, because the act of typing the
;  503          0825  2 ! "A" will cancel the original call.  Second, why the hell would people call
;  504          0826  2 ! themselves if they didn't want to answer the phone?  That's like arguing
;  505          0827  2 ! with yourself and losing the argument!
;  506          0828  2
;  507          0829  2 op = .phn$gq_pubhead[0];
```

```
;  508          0830  3 if phn$cmp_target(caller_tsb,op[pub_b_tsb]) then (
;  509          0831  4          if .first_ring_flag then (
;  510          0832  4                  op[pub_v_answering] = true;
;  511          0833  4                  phn$answer_cmd();
;  512          0834  3          );
;  513          0835  3          return;
;  514          0836  2 );
```

```
;  516          0837  2 ! Someone else is calling us.  This is rather complicated, because there are
;  517          0838  2 ! really 9 distinct cases.  The first case is when we're calling someone else
;  518          0839  2 ! and our phone is busy.  The other 8 cases are all combinations of the
;  519          0840  2 ! following boolean variables:  whether we are currently answering the phone,
;  520          0841  2 ! whether this is a first-time ring, and whether this caller is the same as
;  521          0842  2 ! the person who rang us last time.
;  522          0843  2
;  523          0844  2 if .op[pub_v_calling] or
;  524          0845  3    (.op[pub_v_answering] and .first_ring_flag) then (
;  525          0846  3
;  526          0847  3         ! Either we're placing a call now, or we're already answering the
;  527          0848  3         ! phone and someone else sends us a first-time ring.  In this case we
;  528          0849  3         ! make a temporary link to the caller and tell them we're busy.
;  529          0850  3
;  530          0851  3         status = phn$estab_link(message_dsc,tp);
;  531          0852  3         check (.status);
;  532          0853  3         status = phn$break_link(.tp,smb__busy);
;  533          0854  3         check (.status);
;  534          0855  3         return;
;  535          0856  2 );
;  536          0857  2
;  537          0858  3 if not .op[pub_v_answering] and .first_ring_flag then (
;  538          0859  3
;  539          0860  3         ! We're not answering and it's a first-time ring.  We have a new
;  540          0861  3         ! caller, so establish a link to them.  Set up our PUB so we'll
;  541          0862  3         ! know who's calling.  Display a ring message to the user.
;  542          0863  3
;  543          0864  3         status = phn$estab_link(message_dsc,tp);
;  544          0865  3         check (.status);
;  545          0866  3         op[pub_v_answering] = true;
;  546          0867  3         op[pub_l_busylink] = .tp;
;  547          0868  3         phn$jangle_string(.tp);
;  548          0869  3         return;
;  549          0870  2 );
;  550          0871  2
;  551          0872  2 if (tp = .op[pub_l_busylink]) neqa 0 then
;  552          0873  2         if .op[pub_v_answering] and not .first_ring_flag and
;  553          0874  3            phn$cmp_target(caller_tsb,tp[pub_b_tsb])          then (
;  554          0875  3
;  555          0876  3              ! We're answering, it's a subsequent ring, and it's from the
;  556          0877  3              ! same person.  Again display a ring message to the user.
;  557          0878  3
;  558          0879  3              phn$jangle_string(.tp);
;  559          0880  3              return;
;  560          0881  2         );
;  561          0882  2
;  562          0883  2 ! What the hell?!??!  Can't imagine what's going on at this point, so
;  563          0884  2 ! just ignore the message.
;  564          0885  2
;  565          0886  2 return;
;  566          0887  2
;  567          0888  1 end;
```

```
                                00FC 00000          .ENTRY  PHN$RANG_IN, Save R2,R3,R4,R5,R6,R7     ; 0801
                     57 00000000G   00  9E 00002    MOVAB   LIB$SIGNAL, R7                         ;
                     5E       FF18  CE  9E 00009    MOVAB   -232(SP), SP                           ;
                     53         04  AC  D0 0000E    MOVL    MESSAGE, R3                            ; 0804
                     50             63  3C 00012    MOVZWL  (R3), R0                               ; 0817
                     50         04  A3  C0 00015    ADDL2   4(R3), R0                              ;
                     56         FF  A0  90 00019    MOVB    -1(R0), FIRST_RING_FLAG                ;
                     63             02  A2 0001D    SUBW2   #2, (R3)                               ; 0818
                                04  AE  9F 00020    PUSHAB  CALLER_TSB                             ; 0819
                                    53  DD 00023    PUSHL   R3                                     ;
                0000G  CF           02  FB 00025    CALLS   #2, PHN$MAKE_TSB                       ;
                     55             50  D0 0002A    MOVL    R0, STATUS                             ;
                     05             55  E8 0002D    BLBS    STATUS, 1$                             ; 0820
                                    55  DD 00030    PUSHL   STATUS                                 ;
                     67             01  FB 00032    CALLS   #1, LIB$SIGNAL                         ;
                     52       0000G CF  D0 00035 1$: MOVL    PHN$GQ_PUBHEAD, OP                     ; 0829
                                   0C   A2 9F 0003A    PUSHAB  12(OP)                             ; 0830
                                   08   AE 9F 0003D    PUSHAB  CALLER_TSB                         ;
                0000G  CF           02  FB 00040    CALLS   #2, PHN$CMP_TARGET                     ;
                     0F             50  E9 00045    BLBC    R0, 3$                                 ;
                     01             56  E8 00048    BLBS    FIRST_RING_FLAG, 2$                    ; 0831
                                    04  00 0004B    RET                                            ;
                00F0 C2             10  88 0004C 2$: BISB2   #16, 240(OP)                          ; 0832
                0000V  CF           00  FB 00051    CALLS   #0, PHN$ANSWER_CMD                     ; 0833
                                    04  00 00056    RET                                            ; 0830
                     54       00F0  C2  9E 00057 3$: MOVAB   240(OP), R4                            ; 0844
          07         64             03  E0 0005C    BBS     #3, (R4), 4$                           ;
          31         64             04  E1 00060    BBC     #4, (R4), 7$                           ; 0845
                     2A             56  E9 00064    BLBC    FIRST_RING_FLAG, 6$                    ;
                           4008     8F  BB 00067 4$: PUSHR   #^M<R3,SP>                            ; 0851
                0000G  CF           02  FB 0006B    CALLS   #2, PHN$ESTAB_LINK                     ;
                     55             50  D0 00070    MOVL    R0, STATUS                             ;
                     05             55  E8 00073    BLBS    STATUS, 5$                             ; 0852
                                    55  DD 00076    PUSHL   STATUS                                 ;
                     67             01  FB 00078    CALLS   #1, LIB$SIGNAL                         ;
                                    0A  DD 0007B 5$: PUSHL   #10                                   ; 0853
                                04  AE  DD 0007D    PUSHL   TP                                     ;
                0000G  CF           02  FB 00080    CALLS   #2, PHN$BREAK_LINK                     ;
                     55             50  D0 00085    MOVL    R0, STATUS                             ;
                     4F             55  E8 00088    BLBS    STATUS, 11$                            ; 0854
                                    55  DD 0008B    PUSHL   STATUS                                 ;
                     67             01  FB 0008D    CALLS   #1, LIB$SIGNAL                         ;
                                    04  00 00090    RET                                            ; 0845
          21         64             04  E0 00091 6$: BBS     #4, (R4), 9$                           ; 0858
                     1E             56  E9 00095 7$: BLBC    FIRST_RING_FLAG, 9$                    ;
                           4008     8F  BB 00098    PUSHR   #^M<R3,SP>                             ; 0864
                0000G  CF           02  FB 0009C    CALLS   #2, PHN$ESTAB_LINK                     ;
                     55             50  D0 000A1    MOVL    R0, STATUS                             ;
                     05             55  E8 000A4    BLBS    STATUS, 8$                             ; 0865
                                    55  DD 000A7    PUSHL   STATUS                                 ;
                     67             01  FB 000A9    CALLS   #1, LIB$SIGNAL                         ;
                     64             10  88 000AC 8$: BISB2   #16, (R4)                            ; 0866
                00F8 C2             6E  D0 000AF    MOVL    TP, 248(OP)                            ; 0867
                                    1D  11 000B4    BRB     10$                                    ; 0868
                     6E       00F8  C2  D0 000B6 9$: MOVL    248(OP), TP                            ; 0872
                                    1D  13 000BB    BEQL    11$                                    ;
          19         64             04  E1 000BD    BBC     #4, (R4), 11$                          ; 0873
```

```
                         16              56  E8 000C1          BLBS    FIRST_RING_FLAG, 11$                    ;
                      7E 6E              0C  C1 000C4          ADDL3   #12, TP, -(SP)                        ; 0874
                               08        AE  9F 000C8          PUSHAB  CALLER_FSB                              ;
                   0000G  CF             02  FB 000CB          CALLS   #2, PHN$CMP_TARGET                      ;
                          07             50  E9 000D0          BLBC    R0, 11$                                 ;
                                         6E  DD 000D3 10$:     PUSHL   TP                                      ; 0879
                   0000V  CF             01  FB 000D5          CALLS   #1, PHN$JANGLE_STRING                   ;
                                            04 000DA 11$:      RET                                             ; 0888

; Routine Size:  219 bytes,    Routine Base:  $CODE$ + 027D
```

```
;  569          0889  1 %sbttl 'PHN$JANGLE_STRING - Build and Display Ring Message'
;  570          0890  1 !++
;  571          0891  1 ! Functional Description:
;  572          0892  1 !       This routine is called to build and display a ring message.
;  573          0893  1 !       There are two cases:
;  574          0894  1 !               1.      We are calling someone else and they are not
;  575          0895  1 !                       currently using the PHONE facility.  In this case
;  576          0896  1 !                       we have to broadcast the message to them.
;  577          0897  1 !               2.      Someone is calling us.  In this case we have to
;  578          0898  1 !                       display the message on our message line.
;  579          0899  1 !
;  580          0900  1 ! Formal Parameters:
;  581          0901  1 !       caller_pub      The address of the PUB of the person who is placing
;  582          0902  1 !                       the call.  In case 1, it is our PUB.  In case 2,
;  583          0903  1 !                       it's the PUB of the person calling us.
;  584          0904  1 !       terminal        This argument is only supplied in case 1, and is the
;  585          0905  1 !                       terminal number of the person we are calling.
;  586          0906  1 !
;  587          0907  1 ! Implicit Inputs:
;  588          0908  1 !       global data
;  589          0909  1 !
;  590          0910  1 ! Implicit Outputs:
;  591          0911  1 !       global data
;  592          0912  1 !
;  593          0913  1 ! Returned Value:
;  594          0914  1 !       A boolean, true if we could do it, false if not.
;  595          0915  1 !
;  596          0916  1 ! Side Effects:
;  597          0917  1 !
;  598          0918  1 !--
;  599          0919  1
;  600          0920  1
;  601          0921  2 global routine phn$jangle_string(caller_pub,terminal) = begin
;  602          0922  2
;  603          0923  2 bind
;  604          0924  2         cp = .caller_pub: pub,
;  605          0925  2         caller_tsb = cp[pub_b_tsb]: tsb,
;  606          0926  2         terminal_dsc = .terminal: descriptor;
;  607          0927  2
;  608          0928  2 local
;  609          0929  2         status: long,
;  610          0930  2         brkiosb : vector[4,word],
;  611          0931  2         on_node_length: long;
;  612          0932  2
;  613          0933  2 builtin
;  614          0934  2         nullparameter;
;  615          0935  2
;  616          0936  2
;  617          0937  2 ! We begin by determining if we are on a VAX running DECnet.  If so, we will
;  618          0938  2 ! include the word "on" and the target's node in the ring message.
;  619          0939  2
;  620          0940  2 on_node_length = (if .phn$gq_node_name[len] eqlu 0 then 0 else .phn$gq_node_name[len] + 3);
;  621          0941  2
;  622          0942  2 ! Now we determine if we are to broadcast the ring message.  If so, we get
;  623          0943  2 ! the message text, $fao the various pieces into it, and broadcast it to
;  624          0944  2 ! the terminal.
;  625          0945  2
```

```
;  626          0946  3 if not nullparameter(2) then (
;  627          0947  3         local
;  628          0948  3                 local_described_buffer(msg_buf,79);
;  629          0949  3
;  630          0950  3         ! Get the ring message template.
;  631          0951  3
;  632        P 0952  3         status = $getmsg(msgid=phn$_ring,
;  633        P 0953  3                          msglen=msg_buf[len],
;  634        P 0954  3                          bufadr=msg_buf,
;  635          0955  3                          flags=%b'0001');
;  636          0956  3         check (.status);
;  637          0957  3
;  638          0958  3         ! Build the ring message.
;  639          0959  3
;  640          0960  4         begin
;  641          0961  4         local
;  642          0962  4                 local_described_buffer(fao_buf,79);
;  643          0963  4
;  644        P 0964  4         status = $fao(msg_buf,
;  645        P 0965  4                       fao_buf[len],
;  646        P 0966  4                       fao_buf,
;  647        P 0967  4                       caller_tsb[tsb_q_tkndsc,.caller_tsb[tsb_w_tkncount]-1],
;  648        P 0968  4                       caller_tsb[tsb_q_tkndsc,.caller_tsb[tsb_w_tkncount]],
;  649        P 0969  4                       .on_node_length,
;  650        P 0970  4                       phn$gq_node_name,
;  651          0971  4                       0);
;  652          0972  4         check (.status);
;  653          0973  4
;  654          0974  4         ! Broadcast the ring message to the person's terminal.
;  655          0975  4
;  656        P 0976  4         status = $brkthruw(msgbuf=fao_buf,
;  657        P 0977  4                            sendto=terminal_dsc,
;  658        P 0978  4                            sndtyp=brk$c_device,
;  659        P 0979  4                            reqid=brk$c_phone,
;  660        P 0980  4                            carcon=%C' ',
;  661        P 0981  4                            iosb=brkiosb,
;  662          0982  4                            timout=10);
;  663          0983  4         if .status
;  664          0984  4             then status = .brkiosb[0];
;  665          0985  4         return .status;
;  666          0986  3         end;
;  667          0987  2 );
```

```
;  669      0988  2  ! Well, it appears that we have case 2, and we want to display a ring
;  670      0989  2  ! message to the user.
;  671      0990  2
;  672      0991  2  phn$inform(phn$_ring,
;  673      0992  2             caller_tsb[tsb_q_tkndsc,.caller_tsb[tsb_w_tkncount]-1],
;  674      0993  2             caller_tsb[tsb_q_tkndsc,.caller_tsb[tsb_w_tkncount]],
;  675      0994  2             .on_node_length,
;  676      0995  2             phn$gq_node_name,
;  677      0996  2             0);
;  678      0997  2  return phn$_ok;
;  679      0998  2
;  680      0999  1  end;


                                                        .EXTRN   SYS$GETMSG, SYS$FAO
                                                        .EXTRN   SYS$BRKTHRUW

                                     00FC 00000         .ENTRY   PHN$JANGLE_STRING, Save R2,R3,R4,R5,R6,R7   ; 0921
                    57      0000G  CF  9E 00002         MOVAB    PHN$GQ_NODE_NAME, R7                        ;
                    56  00000000G  00  9E 00007         MOVAB    LIB$SIGNAL, R6                              ;
                    55  00000000G  8F  D0 0000E         MOVL     #PHN$_RING, R5                              ;
                    5E       FF48  CE  9E 00015         MOVAB    -184(SP), SP                                ;
         53       04  AC          0C  C1 0001A          ADDL3    #12, CALLER_PUB, R3                         ; 0925
                    52            67  3C 0001F          MOVZWL   PHN$GQ_NODE_NAME, R2                        ; 0940
                                  04  12 00022          BNEQ     1$                                          ;
                                  52  D4 00024          CLRL     ON_NODE_LENGTH                              ;
                                  03  11 00026          BRB      2$                                          ;
                    52            03  C0 00028 1$:      ADDL2    #3, ON_NODE_LENGTH                          ;
                    02            6C  91 0002B 2$:      CMPB     (AP), #2                                    ; 0946
                                  03  1E 0002E          BGEQU    4$                                          ;
                                008D  31 00030 3$:      BRW      8$                                          ;
                           08     AC  D5 00033 4$:      TSTL     8(AP)                                       ;
                                  F8  13 00036          BEQL     3$                                          ;
              58    AE     4F     8F  9A 00038          MOVZBL   #79, MSG_BUF                                ; 0948
              5C    AE     60     AE  9E 0003D          MOVAB    MSG_BUF+8, MSG_BUF+4                        ;
                    7E            01  7D 00042          MOVQ     #1, -(SP)                                   ; 0955
                           60     AE  9F 00045          PUSHAB   MSG_BUF                                     ;
                           64     AE  9F 00048          PUSHAB   MSG_BUF                                     ;
                                  55  DD 0004B          PUSHL    R5                                          ;
        00000000G   00            05  FB 0004D          CALLS    #5, SYS$GETMSG                              ;
                    54            50  D0 00054          MOVL     R0, STATUS                                  ;
                    05            54  E8 00057          BLBS     STATUS, 5$                                  ; 0956
                                  54  DD 0005A          PUSHL    STATUS                                      ;
                    66            01  FB 0005C          CALLS    #1, LIB$SIGNAL                              ;
                    6E     4F     8F  9A 0005F 5$:      MOVZBL   #79, FAO_BUF                                ; 0962
              04    AE     08     AE  9E 00063          MOVAB    FAO_BUF+8, FAO_BUF+4                        ;
                                  7E  D4 00068          CLRL     -(SP)                                       ; 0971
                         0084     8F  BB 0006A          PUSHR    #^M<R2,R7>                                  ;
                    50     02     A3  3C 0006E          MOVZWL   2(R3), R0                                   ;
                           04   A340  7F 00072          PUSHAQ   4(R3)[R0]                                   ;
                           FC   A340  7F 00076          PUSHAQ   -4(R3)[R0]                                  ;
                           14     AE  9F 0007A          PUSHAB   FAO_BUF                                     ;
                           18     AE  9F 0007D          PUSHAB   FAO_BUF                                     ;
                           74     AE  9F 00080          PUSHAB   MSG_BUF                                     ;
        00000000G   00            08  FB 00083          CALLS    #8, SYS$FAO                                 ;
                    54            50  D0 0008A          MOVL     R0, STATUS                                  ;
```

```
                                05              54  E8 0008D            BLBS    STATUS, 6$                          ; 0972
                                                54  DD 00090            PUSHL   STATUS                              ;
                                66              01  FB 00092            CALLS   #1, LIB$SIGNAL                      ;
                                                7E  7C 00095 6$:        CLRQ    -(SP)                               ; 0982
                                                0A  DD 00097            PUSHL   #10                                 ;
                                                01  DD 00099            PUSHL   #1                                  ;
                                7E                  20  7D 0009B       MOVQ    #32, -(SP)                          ;
                                        F8          AD  9F 0009E        PUSHAB  BRKIOSB                             ;
                                                    01  DD 000A1        PUSHL   #1                                  ;
                                        08          AC  DD 000A3        PUSHL   TERMINAL                            ;
                                        24          AE  9F 000A6        PUSHAB  FAO_BUF                             ;
                                                    7E  D4 000A9        CLRL    -(SP)                               ;
                        00000000G       00              0B  FB 000AB    CALLS   #11, SYS$BRKTHRUW                   ;
                                        54              50  D0 000B2    MOVL    R0, STATUS                          ;
                                        04              54  E9 000B5    BLBC    STATUS, 7$                          ; 0983
                                        54          F8  AD  3C 000B8    MOVZWL  BRKIOSB, STATUS                     ; 0984
                                        50              54  D0 000BC 7$: MOVL    STATUS, R0                          ; 0985
                                                        04  000BF        RET                                         ;
                                                    7E  D4 000C0 8$:    CLRL    -(SP)                               ; 0993
                                                0084    8F  BB 000C2    PUSHR   #^M<R2,R7>                          ; 0994
                                        50          02  A3  3C 000C6    MOVZWL  2(R3), R0                           ; 0993
                                                    04 A340 7F 000CA    PUSHAQ  4(R3)[R0]                           ;
                                                    FC A340 7F 000CE    PUSHAQ  -4(R3)[R0]                          ; 0992
                                                    55  DD 000D2        PUSHL   R5                                  ; 0993
                                0000G   CF              06  FB 000D4    CALLS   #6, PHN$INFORM                      ;
                                        50 00000000G    8F  D0 000D9    MOVL    #PHN$_OK, R0                        ; 0997
                                                        04  000E0        RET                                         ; 0999

; Routine Size:  225 bytes,    Routine Base:  $CODE$ + 0358
```

```
;  682          1000  1 %sbttl 'PHN$ANSWER_CMD - Answer the Phone'
;  683          1001  1 !++
;  684          1002  1 ! Functional Description:
;  685          1003  1 !       This routine handles the ANSWER command, used to answer a call
;  686          1004  1 !       on our phone.  We have to check for various illegal situations,
;  687          1005  1 !       and then we can make the PUB of the caller permanent, and set up
;  688          1006  1 !       a viewport.
;  689          1007  1 !
;  690          1008  1 ! Formal Parameters:
;  691          1009  1 !       none
;  692          1010  1 !
;  693          1011  1 ! Implicit Inputs:
;  694          1012  1 !       global data
;  695          1013  1 !
;  696          1014  1 ! Implicit Outputs:
;  697          1015  1 !       global data
;  698          1016  1 !
;  699          1017  1 ! Returned Value:
;  700          1018  1 !       none
;  701          1019  1 !
;  702          1020  1 ! Side Effects:                                                        ;
;  703          1021  1 !
;  704          1022  1 !--
;  705          1023  1
;  706          1024  1
;  707          1025  2 global routine phn$answer_cmd: novalue = begin
;  708          1026  2
;  709          1027  2 local
;  710          1028  2         status: long,
;  711          1029  2         op: ref pub,                    ! Pointer to our PUB.
;  712          1030  2         cp: ref pub;                    ! Pointer to caller's PUB.
;  713          1031  2
;  714          1032  2
;  715          1033  2 ! If the answering flag in our PUB is not set, then the user has entered
;  716          1034  2 ! a spurious ANSWER command.
;  717          1035  2
;  718          1036  2 op = .phn$gq_pubhead[0];
;  719          1037  3 if not .op[pub_v_answering] or (.op[pub_l_busylink] eqla 0) then (
;  720          1038  3         phn$inform(phn$_nocall);
;  721          1039  3         return;
;  722          1040  2 );
;  723          1041  2
;  724          1042  2 ! Yes, we can answer the phone.  Clear the answering flag and the busylink
;  725          1043  2 ! in our PUB, since we will now have answered.
;  726          1044  2
;  727          1045  2 op[pub_v_answering] = false;
;  728          1046  2 cp = .op[pub_l_busylink];
;  729          1047  2 op[pub_l_busylink] = 0;
;  730          1048  2
;  731          1049  2 ! Now we have to see if we are about to establish a redundant conversation,
;  732          1050  2 ! that is, a conversation with someone we are already talking to.  The
;  733          1051  2 ! caller should have prevented this, but let's hear it for robustness!
;  734          1052  2
;  735          1053  3 if phn$redundant(.cp) then (
;  736          1054  3         phn$break_link(.cp,smb__busy);
;  737          1055  3         phn$inform(phn$_ivredunans);
;  738          1056  3         return;
```

```
;  739          1057  2 );
;  740          1058  2
;  741          1059  2 ! OK, now we can send an answered message to the caller so they'll know
;  742          1060  2 ! that we have answered their call.
;  743          1061  2
;  744          1062  2 phn$send_smb(.cp,smb__answered);
;  745          1063  2
;  746          1064  2 ! Now we see if the call we are answering is from ourselves.  If so,
;  747          1065  2 ! we're done.  The PUB will be made permanent by PHN$ANSWERED when it
;  748          1066  2 ! receives our answered message.  But we must re-establish the busylink.
;  749          1067  2
;  750          1068  3 if phn$cmp_target(cp[pub_b_tsb],op[pub_b_tsb]) then (
;  751          1069  3         op[pub_l_busylink] = .cp;
;  752          1070  3         return;
;  753          1071  2 );
;  754          1072  2
;  755          1073  2 ! We are answering a call from someone else.  In order to be able to
;  756          1074  2 ! talk, we have to make their PUB permanent and assign them a viewport.
;  757          1075  2
;  758          1076  2 cp[pub_v_temporary] = false;
;  759          1077  2 status = phn$fresh_screen(false);
;  760          1078  3 if .status nequ phn$_ok then (
;  761          1079  3         phn$break_link(.cp,smb__busy);
;  762          1080  3         phn$inform(.status);
;  763          1081  2 );
;  764          1082  2 return;
;  765          1083  2
;  766          1084  1 end;
```

```
                                                001C 00000          .ENTRY  PHN$ANSWER_CMD, Save R2,R3,R4                   ; 1025
                    52      0000G   CF   D0 00002          MOVL    PHN$GQ_PUBREAD, OP                              ; 1036
06      00F0        C2              04   E1 00007          BBC     #4, 240(OP), 1$                                 ; 1037
                            00F8    C2   D5 0000D          TSTL    248(OP)
                                    08   12 00011          BNEQ    2$
                        00000000G   8F   DD 00013 1$:      PUSHL   #PHN$_NOCALL                                    ; 1038
                                    68   11 00019          BRB     5$
                00F0    C2          10   8A 0001B 2$:      BICB2   #16, 240(OP)                                    ; 1045
                        54      00F8    C2   9E 00020      MOVAB   248(OP), R4                                     ; 1046
                        53              64   D0 00025      MOVL    (R4), CP
                                        64   D4 00028      CLRL    (R4)                                            ; 1047
                                        53   DD 0002A      PUSHL   CP                                              ; 1053
                0000G   CF              01   FB 0002C      CALLS   #1, PHN$REDUNDANT
                        11              50   E9 00031      BLBC    R0, 3$
                                        0A   DD 00034      PUSHL   #10                                             ; 1054
                                        53   DD 00036      PUSHL   CP
                0000G   CF              02   FB 00038      CALLS   #2, PHN$BREAK_LINK
                        00000000G       8F   DD 0003D      PUSHL   #PHN$_IVREDUNANS                                ; 1055
                                        3E   11 00043      BRB     5$
                                        0B   DD 00045 3$:   PUSHL   #11                                             ; 1062
                                        53   DD 00047      PUSHL   CP
                0000G   CF              02   FB 00049      CALLS   #2, PHN$SEND_SMB
                                0C      A2   9F 0004E      PUSHAB  12(OP)                                          ; 1068
                                0C      A3   9F 00051      PUSHAB  12(CP)
```

```
                        0000G  CF            02  FB 00054          CALLS   #2, PHN$CMP_TARGET                ;
                               04            50  E9 00059          BLBC    R0, 4$                           ;
                               64            53  D0 0005C          MOVL    CP, (R4)                         ; 1069
                                             04 0005F             RET                                      ; 1068
                        00F0   C3            04  8A 00060 4$:     BICB2   #4, 240(CP)                      ; 1076
                                             7E  D4 00065         CLRL    -(SP)                            ; 1077
                        0000G  CF            01  FB 00067         CALLS   #1, PHN$FRESH_SCREEN             ;
                               52            50  D0 0006C         MOVL    R0, STATUS                       ;
                    00000000G  8F            52  D1 0006F         CMPL    STATUS, #PHN$_OK                 ; 1078
                                             10  13 00076         BEQL    6$                               ;
                                             0A  DD 00078         PUSHL   #10                              ; 1079
                                             53  DD 0007A         PUSHL   CP                               ;
                        0000G  CF            02  FB 0007C         CALLS   #2, PHN$BREAK_LINK               ;
                                             52  DD 00081         PUSHL   STATUS                           ; 1080
                        0000G  CF            01  FB 00083 5$:     CALLS   #1, PHN$INFORM                   ;
                                                 04 00088 6$:     RET                                      ; 1084

; Routine Size:  137 bytes,    Routine Base:  $CODE$ + 0439
```

```
;  768          1085  1 %sbttl 'PHN$ANSWERED - Handle Someone Answering Us'
;  769          1086  1 !++
;  770          1087  1 ! Functional Description:
;  771          1088  1 !       This steering message routine handles the answered message, which
;  772          1089  1 !       is set to us when someone answers a call we placed.  This message
;  773          1090  1 !       forces us to make their PUB permanent and assign a viewport.
;  774          1091  1 !
;  775          1092  1 ! Formal Parameters:
;  776          1093  1 !       from_msg        Address of descriptor of the node/user name spec
;  777          1094  1 !                       of the person who sent the message.  It is followed
;  778          1095  1 !                       by an eofrom character.
;  779          1096  1 !
;  780          1097  1 ! Implicit Inputs:
;  781          1098  1 !       global data
;  782          1099  1 !
;  783          1100  1 ! Implicit Outputs:
;  784          1101  1 !       global data
;  785          1102  1 !
;  786          1103  1 ! Returned Value:
;  787          1104  1 !       none
;  788          1105  1 !
;  789          1106  1 ! Side Effects:
;  790          1107  1 !
;  791          1108  1 !--
;  792          1109  1
;  793          1110  1
;  794          1111  2 global routine phn$answered(from_msg): novalue = begin
;  795          1112  2
;  796          1113  2 bind
;  797          1114  2         from_msg_dsc = .from_msg: descriptor;
;  798          1115  2
;  799          1116  2 local
;  800          1117  2         status: long,
;  801          1118  2         op: ref pub,                    ! Pointer to our PUB.
;  802          1119  2         spec_tsb: tsb,                  ! TSB to parse answerer's spec.
;  803          1120  2         tp: ref pub;                    ! Pointer to answerer's PUB.
;  804          1121  2
;  805          1122  2
;  806          1123  2 ! We begin by ensuring that we are actually calling someone.  If not,
;  807          1124  2 ! it's a spurious message.
;  808          1125  2
;  809          1126  2 op = .phn$gq_pubhead[0];
;  810          1127  2 if not .op[pub_v_calling] or (.op[pub_l_busylink] eqla 0) then
;  811          1128  2         return;
;  812          1129  2
;  813          1130  2 ! Now we will create a TSB so we can parse the spec of the person answering.
;  814          1131  2 ! We need to ensure that it really is the person we are calling.
;  815          1132  2
;  816          1133  2 dec (from_msg_dsc[len]);
;  817          1134  2 status = phn$make_tsb(from_msg_dsc,spec_tsb);
;  818          1135  2 check (.status);
;  819          1136  2 tp = .op[pub_l_busylink];
;  820          1137  2 if not phn$cmp_target(spec_tsb,tp[pub_b_tsb]) then
;  821          1138  2         return;
;  822          1139  2
;  823          1140  2 ! It appears that the call is complete.  Clean up calling info in our PUB.
;  824          1141  2
```

```
;  825          1142  2 op[pub_v_calling] = false;
;  826          1143  2 op[pub_l_busylink] = 0;
;  827          1144  2
;  828          1145  2 ! Now we can make the person's PUB permanent and assign them a viewport.
;  829          1146  2
;  830          1147  2 tp[pub_v_temporary] = false;
;  831          1148  2 status = phn$fresh_screen(false);
;  832          1149  3 if .status nequ phn$_ok then (
;  833          1150  3         phn$break_link(.tp,smb__hungup);
;  834          1151  3         phn$inform(.status);
;  835          1152  3         return;
;  836          1153  2 );
;  837          1154  2
;  838          1155  2 ! Last but not least, we have to force links between the new person and
;  839          1156  2 ! anyone else we might be talking to, creating a conference call.
;  840          1157  2 ! Finally!  We're ready to talk to the person.
;  841          1158  2
;  842          1159  2 phn$force_links(.tp);
;  843          1160  2 phn$inform(phn$_answered);
;  844          1161  2 return;
;  845          1162  2
;  846          1163  1 end;


                                 001C 00000          .ENTRY  PHN$ANSWERED, Save R2,R3,R4                     ; 1111
                          5E     FF1C CE   9E 00002   MOVAB   -228(SP), SP                                    ;
                          53     0000G CF   D0 00007   MOVL    PHN$GQ_PUBHEAD, OP                            ; 1126
                75        00F0   C3       03   E1 0000C   BBC     #3, 240(OP), 4$                           ; 1127
                                 00F8   C3   D5 00012   TSTL    248(OP)                                     ;
                                        6F   13 00016   BEQL    4$                                          ;
                                          04 BC   B7 00018   DECW    @FROM_MSG                               ; 1133
                                        5E   DD 0001B   PUSHL   SP                                          ; 1134
                                          04 AC   DD 0001D   PUSHL   FROM_MSG                               ;
                                0000G   CF       02   FB 00020   CALLS   #2, PHN$MAKE_TSB                  ;
                                        54       50   D0 00025   MOVL    R0, STATUS                         ;
                                        09       54   E8 00028   BLBS    STATUS, 1$                         ; 1135
                                                 54   DD 0002B   PUSHL   STATUS                             ;
                            00000000G   00       01   FB 0002D   CALLS   #1, LIB$SIGNAL                     ;
                                        52 00F8  C3   D0 00034 1$: MOVL    248(OP), TP                        ; 1136
                                          0C     A2   9F 00039   PUSHAB  12(TP)                             ; 1137
                                          04     AE   9F 0003C   PUSHAB  SPEC_TSB                           ;
                                0000G   CF       02   FB 0003F   CALLS   #2, PHN$CMP_TARGET                 ;
                                        40       50   E9 00044   BLBC    R0, 4$                             ;
                                00F0    C3       08   8A 00047   BICB2   #8, 240(OP)                        ; 1142
                                        00F8 C3   D4 0004C   CLRL    248(OP)                                ; 1143
                                00F0    C2       04   8A 00050   BICB2   #4, 240(TP)                        ; 1147
                                        7E       D4   00055   CLRL    -(SP)                              ; 1148
                                0000G   CF       01   FB 00057   CALLS   #1, PHN$FRESH_SCREEN               ;
                                        54       50   D0 0005C   MOVL    R0, STATUS                         ;
                            00000000G   8F       54   D1 0005F   CMPL    STATUS, #PHN$_OK                   ; 1149
                                                 0D   13 00066   BEQL    2$                                 ;
                                                 09   DD 00068   PUSHL   #9                                 ; 1150
                                                 52   DD 0006A   PUSHL   TP                                  ;
                                0000G   CF       02   FB 0006C   CALLS   #2, PHN$BREAK_LINK                 ;
```

```
                                                54  DD 00071          PUSHL   STATUS                                ; 1151
                                                0D  11 00073          BRB     3$
                                                52  DD 00075 2$:      PUSHL   TP                                  ; 1159
                                      0000G     CF  01  FB 00077          CALLS   #1, PHN$FORCE_LINKS
                                                    8F  DD 0007C          PUSHL   #PHN$_ANSWERED                      ; 1160
                                      0000G     CF  01  FB 00082 3$:      CALLS   #1, PHN$INFORM
                                                    04 00087 4$:      RET                                         ; 1163

; Routine Size:  136 bytes,    Routine Base:  $CODE$ + 04C2
```

```
;  848          1164  1 %sbttl 'PHN$REJECT_CMD - Handle REJECT Command'
;  849          1165  1 !++
;  850          1166  1 ! Functional Description:
;  851          1167  1 !       This routine handles the REJECT command, which the user enters
;  852          1168  1 !       to reject a phone call.  We have to send the caller a rejected
;  853          1169  1 !       message so the ringing will stop.
;  854          1170  1 !
;  855          1171  1 ! Formal Parameters:
;  856          1172  1 !       parameter       If the user specifies the optional EXIT parameter,
;  857          1173  1 !                       we will exit after the rejection.
;  858          1174  1 !
;  859          1175  1 ! Implicit Inputs:
;  860          1176  1 !       global data
;  861          1177  1 !
;  862          1178  1 ! Implicit Outputs:
;  863          1179  1 !       global data
;  864          1180  1 !
;  865          1181  1 ! Returned Value:
;  866          1182  1 !       none
;  867          1183  1 !
;  868          1184  1 ! Side Effects:
;  869          1185  1 !
;  870          1186  1 !--
;  871          1187  1
;  872          1188  1
;  873          1189  2 global routine phn$reject_cmd(parameter): novalue = begin
;  874          1190  2
;  875          1191  2 bind
;  876          1192  2         parameter_dsc = .parameter: descriptor;
;  877          1193  2
;  878          1194  2 local
;  879          1195  2         op: ref pub,                    ! Pointer to our PUB.
;  880          1196  2         cp: ref pub;                    ! Pointer to caller's PUB.
;  881          1197  2
;  882          1198  2
;  883          1199  2 ! First we must ensure that someone is actually calling us.  If not, the
;  884          1200  2 ! user must be confused.
;  885          1201  2
;  886          1202  2 op = .phn$gq_pubhead[0];
;  887          1203  3 if not .op[pub_v_answering] or (.op[pub_l_busylink] eqla 0) then (
;  888          1204  3         phn$inform(phn$_nocall);
;  889          1205  3         return;
;  890          1206  2 );
;  891          1207  2
;  892          1208  2 ! Now we can break the link to the caller, sending them a rejected message.
;  893          1209  2
;  894          1210  2 op[pub_v_answering] = false;
;  895          1211  2 cp = .op[pub_l_busylink];
;  896          1212  2 op[pub_l_busylink] = 0;
;  897          1213  2 phn$break_link(.cp,smb__rejected);
;  898          1214  2
;  899          1215  2 ! Now we will check for the EXIT parameter.  If present, we will force
;  900          1216  2 ! an exit command.  Garbage will cause an error message.
;  901          1217  2
;  902          1218  3 if .parameter_dsc[len] gtru 0 then (
;  903          1219  3         if ch$eql(.parameter_dsc[len],.parameter_dsc[ptr],
;  904          1220  3                 minu(.parameter_dsc[len],4),uplit byte ('EXIT'),' ') then
```

```
;  905          1221  3                    phn$exit_cmd()
;  906          1222  3            else
;  907          1223  3                    phn$inform(phn$_rejectjunk,parameter_dsc);
;  908          1224  2 );
;  909          1225  2
;  910          1226  2 return;
;  911          1227  2
;  912          1228  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    54  49  58  45  0006D P.AAI:  .ASCII  \EXIT\                                          ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                  001C 00000    .ENTRY  PHN$REJECT_CMD, Save R2,R3,R4                   ; 1189
                                54        04      AC   D0 00002 MOVL    PARAMETER, R4                                   ; 1192
                                52      0000G     CF   D0 00006 MOVL    PHN$GQ_PUBHEAD, OP                              ; 1202
                    06      00F0    C2          04   E1 0000B BBC     #4, 240(OP), 1$                                 ; 1203
                                          00F8  C2   D5 00011 TSTL    248(OP)                                         ;
                                                0C   12 00015 BNEQ    2$                                              ;
                                   00000000G    8F   DD 00017 1$:     PUSHL   #PHN$_NOCALL                            ; 1204
                                0000G   CF              01   FB 0001D CALLS   #1, PHN$INFORM                          ;
                                                        04 00022      RET                                             ; 1203
                                00F0    C2              10   8A 00023 2$:     BICB2   #16, 240(OP)                    ; 1210
                                        50        00F8  C2   D0 00028 MOVL    248(OP), CP                             ; 1211
                                                  00F8  C2   D4 0002D CLRL    248(OP)                                 ; 1212
                                                        0C   DD 00031 PUSHL   #12                                     ; 1213
                                                        50   DD 00033 PUSHL   CP                                      ;
                                0000G   CF              02   FB 00035 CALLS   #2, PHN$BREAK_LINK                      ;
                                                        64   B5 0003A TSTW    (R4)                                    ; 1218
                                                        29   13 0003C BEQL    5$                                      ;
                                        50              64   3C 0003E MOVZWL  (R4), R0                                ; 1220
                                        04              50   B1 00041 CMPW    R0, #4                                  ;
                                                        03   1B 00044 BLEQU   3$                                      ;
                                        50              04   D0 00046 MOVL    #4, R0                                  ;
          50            20      04    B4              64   2D 00049 3$:     CMPC5   (R4), @4(R4), #32, R0, P.AAI    ; 1219
                                              0000'     CF      0004F                                                   ;
                                                        06   12 00052 BNEQ    4$                                      ;
                                0000G   CF              00   FB 00054 CALLS   #0, PHN$EXIT_CMD                        ; 1221
                                                        04 00059      RET                                             ;
                                                        54   DD 0005A 4$:     PUSHL   R4                              ; 1223
                                   00000000G            8F   DD 0005C PUSHL   #PHN$_REJECTJUNK                       ;
                                0000G   CF              02   FB 00062 CALLS   #2, PHN$INFORM                          ;
                                                        04 00067 5$:     RET                                          ; 1228

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 054A
```

```
;  914          1229  1 %sbttl 'PHN$REJECTED - Handle Rejected Message'
;  915          1230  1 !++
;  916          1231  1 ! Functional Description:
;  917          1232  1 !       This steering message routine handles the rejected message,
;  918          1233  1 !       which someone sends us when they reject our phone call.
;  919          1234  1 !       This message forces us to cancel the call.
;  920          1235  1 !
;  921          1236  1 ! Formal Parameters:
;  922          1237  1 !       from_msg        The address of the descriptor of the node/user name
;  923          1238  1 !                       spec of the person rejecting our call.  It is
;  924          1239  1 !                       followed by an eofrom character.
;  925          1240  1 !
;  926          1241  1 ! Implicit Inputs:
;  927          1242  1 !       global data
;  928          1243  1 !
;  929          1244  1 ! Implicit Outputs:
;  930          1245  1 !       global data
;  931          1246  1 !
;  932          1247  1 ! Returned Value:
;  933          1248  1 !       none
;  934          1249  1 !
;  935          1250  1 ! Side Effects:
;  936          1251  1 !
;  937          1252  1 !--
;  938          1253  1
;  939          1254  1
;  940          1255  2 global routine phn$rejected(from_msg): novalue = begin
;  941          1256  2
;  942          1257  2 bind
;  943          1258  2         from_msg_dsc = .from_msg: descriptor;
;  944          1259  2
;  945          1260  2 local
;  946          1261  2         status: long,
;  947          1262  2         op: ref pub,                    ! Pointer to our PUB.
;  948          1263  2         spec_tsb: tsb,                  ! TSB to parse answerer's spec.
;  949          1264  2         tp: ref pub;                    ! Pointer to answerer's PUB.
;  950          1265  2
;  951          1266  2
;  952          1267  2 ! We begin by ensuring that we are actually calling someone.  If not,
;  953          1268  2 ! it's a spurious message.
;  954          1269  2
;  955          1270  2 op = .phn$gq_pubhead[0];
;  956          1271  2 if not .op[pub_v_calling] or (.op[pub_l_busylink] eqla 0) then
;  957          1272  2         return;
;  958          1273  2
;  959          1274  2 ! Now we will create a TSB so we can parse the spec of the person rejecting.
;  960          1275  2 ! We need to ensure that it really is the person we are calling.
;  961          1276  2
;  962          1277  2 dec (from_msg_dsc[len]);
;  963          1278  2 status = phn$make_tsb(from_msg_dsc,spec_tsb);
;  964          1279  2 check (.status);
;  965          1280  2 tp = .op[pub_l_busylink];
;  966          1281  2 if not phn$cmp_target(spec_tsb,tp[pub_b_tsb]) then
;  967          1282  2         return;
;  968          1283  2
;  969          1284  2 ! Now we can break the call to the person.  Also inform the user.
;  970          1285  2
```

```
;  971          1286  2 phn$break_call();
;  972          1287  2 phn$inform(phn$_rejected);
;  973          1288  2 return;
;  974          1289  2
;  975          1290  1 end;


                                                    0004 00000          .ENTRY  PHN$REJECTED, Save R2                   ; 1255
                                                5E  FF1C CE  9E 00002   MOVAB   -228(SP), SP                           ;
                                                52  0000G CF  D0 00007   MOVL    PHN$GQ_PUBHEAD, OP                      ; 1270
                                            42      00F0 C2  03 E1 0000C   BBC     #3, 240(OP), 2$                         ; 1271
                                                         00F8 C2  D5 00012   TSTL    248(OP)                                 ;
                                                              3C  13 00016   BEQL    2$                                      ;
                                                           04   BC  B7 00018   DECW    @FROM_MSG                               ; 1277
                                                              5E  DD 0001B   PUSHL   SP                                      ; 1278
                                                           04   AC  DD 0001D   PUSHL   FROM_MSG                             ;
                                                    0000G  CF      02  FB 00020   CALLS   #2, PHN$MAKE_TSB                    ;
                                                        09         50  E8 00025   BLBS    STATUS, 1$                          ; 1279
                                                                   50  DD 00028   PUSHL   STATUS                              ;
                                                00000000G  00      01  FB 0002A   CALLS   #1, LIB$SIGNAL                     ;
                                                        50  00F8   C2  D0 00031 1$: MOVL    248(OP), TP                         ; 1280
                                                            0C     A0  9F 00036   PUSHAB  12(TP)                              ; 1281
                                                            04     AE  9F 00039   PUSHAB  SPEC_TSB                            ;
                                                    0000G  CF      02  FB 0003C   CALLS   #2, PHN$CMP_TARGET               ;
                                                        10         50  E9 00041   BLBC    R0, 2$                              ;
                                                    0000G  CF      00  FB 00044   CALLS   #0, PHN$BREAK_CALL               ; 1286
                                                        00000000G  8F  DD 00049   PUSHL   #PHN$_REJECTED                   ; 1287
                                                    0000G  CF      01  FB 0004F   CALLS   #1, PHN$INFORM                   ;
                                                                   04  00054 2$:  RET                                     ; 1290

; Routine Size:  85 bytes,    Routine Base:  $CODE$ + 05B2
```

```
;  977     1291  1 %sbttl 'PHN$BUSY - Handle Busy Message'
;  978     1292  1 !++
;  979     1293  1 ! Functional Description:
;  980     1294  1 !       This steering message routine handles the busy message, which
;  981     1295  1 !       someone sends us when they are busy and can't continue.
;  982     1296  1 !       This message usually comes from someone we are calling, but in
;  983     1297  1 !       very bizarre circumstances can also come from someone calling us.
;  984     1298  1 !
;  985     1299  1 ! Formal Parameters:
;  986     1300  1 !       from_msg        The address of the descriptor of the node/user name
;  987     1301  1 !                       spec of the person who is busy.  It is followed by
;  988     1302  1 !                       an eofrom character.
;  989     1303  1 !
;  990     1304  1 ! Implicit Inputs:
;  991     1305  1 !       global data
;  992     1306  1 !
;  993     1307  1 ! Implicit Outputs:
;  994     1308  1 !       global data
;  995     1309  1 !
;  996     1310  1 ! Returned Value:
;  997     1311  1 !       none
;  998     1312  1 !
;  999     1313  1 ! Side Effects:
; 1000     1314  1 !
; 1001     1315  1 !--
; 1002     1316  1
; 1003     1317  1
; 1004     1318  2 global routine phn$busy(from_msg): novalue = begin
; 1005     1319  2
; 1006     1320  2 bind
; 1007     1321  2         from_msg_dsc = .from_msg: descriptor;
; 1008     1322  2
; 1009     1323  2 local
; 1010     1324  2         status: long,
; 1011     1325  2         op: ref pub,                    ! Points to our PUB.
; 1012     1326  2         spec_tsb. tsb,                  ! TSB to parse busy person's spec.
; 1013     1327  2         tp: ref pub;                    ! Points to busy person's PUB.
; 1014     1328  2
; 1015     1329  2
; 1016     1330  2 ! First we ensure that we are either calling or answering someone.  If not
; 1017     1331  2 ! ignore the message.
; 1018     1332  2
; 1019     1333  2 op = .phn$gq_pubhead[0];
; 1020     1334  2 if .op[pub_l_busylink] eqla 0 then
; 1021     1335  2         return;
; 1022     1336  2
; 1023     1337  2 ! Now we will create a TSB so we can parse the spec of the busy person.
; 1024     1338  2 ! We need to ensure that it really is the person we are calling/answering.
; 1025     1339  2
; 1026     1340  2 dec (from_msg_dsc[len]);
; 1027     1341  2 status = phn$make_tsb(from_msg_dsc,spec_tsb);
; 1028     1342  2 check (.status);
; 1029     1343  2 tp = .op[pub_l_busylink];
; 1030     1344  2 if not phn$cmp_target(spec_tsb,tp[pub_b_tsb]) then
; 1031     1345  2         return;
; 1032     1346  2
; 1033     1347  2 ! Now if we are calling the busy person, we better break the call.
```

```
;  1034          1348  2 ! If they are calling us, just assume something went wrong and forget it.
;  1035          1349  2
;  1036          1350  3 if .op[pub_v_calling] then (
;  1037          1351  3         phn$break_call();
;  1038          1352  3         phn$inform(phn$_busycall);
;  1039          1353  2 );
;  1040          1354  3 if .op[pub_v_answering] then (
;  1041          1355  3         op[pub_v_answering] = false;
;  1042          1356  3         op[pub_l_busylink] = 0;
;  1043          1357  3         phn$break_link(.tp,smb__busy);
;  1044          1358  3         phn$inform(phn$_dead);
;  1045          1359  2 );
;  1046          1360  2 return;
;  1047          1361  2
;  1048          1362  1 end;


                                                001C 00000          .ENTRY  PHN$BUSY, Save R2,R3,R4                 ; 1318
                                          5E    FF1C   CE  9E 00002 MOVAB   -228(SP), SP                            ;
                                          52    0000G  CF  D0 00007 MOVL    PHN$GQ_PUBHEAD, OP                      ; 1333
                                          53    00F8   C2  D0 0000C MOVL    248(OP), R3                             ; 1334
                                                       62  13 00011 BEQL    3$                                      ;
                                                  04   BC  B7 00013 DECW    @FROM_MSG                               ; 1340
                                                       5E  DD 00016 PUSHL   SP                                      ; 1341
                                                  04   AC  DD 00018 PUSHL   FROM_MSG                               ;
                                              0000G CF         02  FB 0001B CALLS   #2, PHN$MAKE_TSB                ;
                                                    09         50  E8 00020 BLBS    STATUS, 1$                      ; 1342
                                                               50  DD 00023 PUSHL   STATUS                          ;
                                          00000000G 00         01  FB 00025 CALLS   #1, LIB$SIGNAL                  ;
                                                    54         53  D0 0002C 1$:    MOVL    R3, TP                  ; 1343
                                                          0C   A4  9F 0002F PUSHAB  12(TP)                         ; 1344
                                                          04   AE  9F 00032 PUSHAB  SPEC_TSB                       ;
                                                   0000G  CF         02  FB 00035 CALLS   #2, PHN$CMP_TARGET         ;
                                                    38         50  E9 0003A BLBC    R0, 3$                          ;
                                                    53    00F0   C2  9E 0003D MOVAB   240(OP), R3                    ; 1350
                                              10    63         03  E1 00042 BBC     #3, (R3), 2$                    ;
                                                   0000G  CF         00  FB 00046 CALLS   #0, PHN$BREAK_CALL         ; 1351
                                                          00000000G  8F  DD 0004B PUSHL   #PHN$_BUSYCALL            ; 1352
                                                   0000G  CF         01  FB 00051 CALLS   #1, PHN$INFORM             ;
                                              1B    63         04  E1 00056 2$:    BBC     #4, (R3), 3$            ; 1354
                                                    63         10  8A 0005A BICB2   #16, (R3)                       ; 1355
                                                          00F8   C2  D4 0005D CLRL    248(OP)                         ; 1356
                                                                 0A  DD 00061 PUSHL   #10                           ; 1357
                                                                 54  DD 00063 PUSHL   TP                            ;
                                                   0000G  CF         02  FB 00065 CALLS   #2, PHN$BREAK_LINK         ;
                                                          00000000G  8F  DD 0006A PUSHL   #PHN$_DEAD                ; 1358
                                                   0000G  CF         01  FB 00070 CALLS   #1, PHN$INFORM             ;
                                                                       04  00075 3$:    RET                         ; 1362

; Routine Size:  118 bytes,    Routine Base:  $CODE$ + 0607
```

```
;  1050          1363  1 %sbttl 'PHN$TALK - Handle Conversation Text We Typed'
;  1051          1364  1 !++
;  1052          1365  1 ! Functional Description:
;  1053          1366  1 !       This steering message routines handles the talk message, which
;  1054          1367  1 !       contains conversation text that we typed.  The text has to be
;  1055          1368  1 !       displayed in our viewport and sent to everyone we are talking
;  1056          1369  1 !       to now.
;  1057          1370  1 !
;  1058          1371  1 ! Formal Parameters:
;  1059          1372  1 !       text            The address of the descriptor of the text we typed.
;  1060          1373  1 !
;  1061          1374  1 ! Implicit Inputs:
;  1062          1375  1 !       global data
;  1063          1376  1 !
;  1064          1377  1 ! Implicit Outputs:
;  1065          1378  1 !       global data
;  1066          1379  1 !
;  1067          1380  1 ! Returned Value:
;  1068          1381  1 !       none
;  1069          1382  1 !
;  1070          1383  1 ! Side Effects:
;  1071          1384  1 !
;  1072          1385  1 !--
;  1073          1386  1
;  1074          1387  1
;  1075          1388  2 global routine phn$talk(text): novalue = begin
;  1076          1389  2
;  1077          1390  2 local
;  1078          1391  2         p: ref pub;
;  1079          1392  2
;  1080          1393  2
;  1081          1394  2 ! We now scan the PUB chain (but not our PUB), looking for people we are
;  1082          1395  2 ! talking to.  We send them the text.
;  1083          1396  2
;  1084          1397  2 p = ..phn$gq_pubhead[0];
;  1085          1398  3 until .p eqla phn$gq_pubhead do (
;  1086          1399  3         if not .p[pub_v_temporary] and
;  1087          1400  3            not .p[pub_v_uhaveheld] and
;  1088          1401  3            not .p[pub_v_hasuheld]       then
;  1089          1402  3
;  1090          1403  3              phn$send_smb(.p,smb__listen,.text);
;  1091          1404  3
;  1092          1405  3         p = .p[pub_l_flink];
;  1093          1406  2 );
;  1094          1407  2
;  1095          1408  2 ! Now we can actually display the text in our own viewport.  This was done
;  1096          1409  2 ! last because I think it gives the user a better feel for response time.
;  1097          1410  2
;  1098          1411  2 phn$show_text(.phn$gq_pubhead[0],.text);
;  1099          1412  2 return;
;  1100          1413  2
;  1101          1414  1 end;
```

```
                                                0004 00000          .ENTRY  PHN$TALK, Save R2                       ; 1388
                                  52    0000G   DF   D0 00002          MOVL    @PHN$GQ_PUBHEAD, P                      ; 1397
                                  50    0000G   CF   9E 00007 1$:      MOVAB   PHN$GQ_PUBHEAD, R0                      ; 1398
                                  50            52   D1 0000C          CMPL    P, R0                                   ;
                                                21   13 0000F          BEQL    3$                                      ;
                                  50    00F0    C2   9E 00011          MOVAB   240(P), R0                              ; 1399
                           13     60            02   E0 00016          BBS     #2, (R0), 2$                            ;
                                  10            60   E8 0001A          BLBS    (R0), 2$                                ; 1400
                           0C     60            01   E0 0001D          BBS     #1, (R0), 2$                            ; 1401
                                          04    AC   DD 00021          PUSHL   TEXT                                    ; 1403
                                                0E   DD 00024          PUSHL   #14                                     ;
                                                52   DD 00026          PUSHL   P                                       ;
                        0000G   CF              03   FB 00028          CALLS   #3, PHN$SEND_SMB                        ;
                                  52            62   D0 0002D 2$:      MOVL    (P), P                                  ; 1405
                                                D5   11 00030          BRB     1$                                      ; 1398
                                          04    AC   DD 00032 3$:      PUSHL   TEXT                                    ; 1411
                                  0000G   CF    DD   00035             PUSHL   PHN$GQ_PUBHEAD                          ;
                        0000G   CF              02   FB 00039          CALLS   #2, PHN$SHOW_TEXT                       ;
                                                04   0003E          RET                                             ; 1414

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 067D
```

```
;  1103          1415  1 %sbttl 'PHN$LISTEN - Handle Conversation Text Someone Else Typed'
;  1104          1416  1 !++
;  1105          1417  1 ! Functional Description:
;  1106          1418  1 !       This steering message routine handles the listen message, which
;  1107          1419  1 !       someone sends us when they type conversation text for us to see.
;  1108          1420  1 !
;  1109          1421  1 ! Formal Parameters:
;  1110          1422  1 !       text_msg        Address of descriptor of two strings, separated
;  1111          1423  1 !                       by an eofrom character:
;  1112          1424  1 !                               1. The node/user name spec of the sender.
;  1113          1425  1 !                               2. The text they want us to see.
;  1114          1426  1 !
;  1115          1427  1 ! Implicit Inputs:
;  1116          1428  1 !       global data
;  1117          1429  1 !
;  1118          1430  1 ! Implicit Outputs:
;  1119          1431  1 !       global data
;  1120          1432  1 !
;  1121          1433  1 ! Returned Value:
;  1122          1434  1 !       none
;  1123          1435  1 !
;  1124          1436  1 ! Side Effects:
;  1125          1437  1 !
;  1126          1438  1 !--
;  1127          1439  1
;  1128          1440  1
;  1129          1441  2 global routine phn$listen(text_msg): novalue = begin
;  1130          1442  2
;  1131          1443  2 bind
;  1132          1444  2         text_msg_dsc = .text_msg: descriptor;
;  1133          1445  2
;  1134          1446  2 local
;  1135          1447  2         status: long,
;  1136          1448  2         text_dsc: descriptor,
;  1137          1449  2         spec_tsb: tsb,
;  1138          1450  2         p: ref pub;
;  1139          1451  2
;  1140          1452  2
;  1141          1453  2 ! We begin by rebuilding the message descriptor so that it only describes
;  1142          1454  2 ! the sender's spec.  We build a second descriptor for the text itself.
;  1143          1455  2
;  1144          1456  2 text_dsc[ptr] = ch$find_ch(.text_msg_dsc[len],.text_msg_dsc[ptr],eofrom) + 1;
;  1145          1457  2 text_dsc[len] = .text_msg_dsc[len] - (.text_dsc[ptr] - .text_msg_dsc[ptr]);
;  1146          1458  2 text_msg_dsc[len] = .text_dsc[ptr] - .text_msg_dsc[ptr] - 1;
;  1147          1459  2
;  1148          1460  2 ! Now we make a TSB to parse the sender's spec.
;  1149          1461  2
;  1150          1462  2 status = phn$make_tsb(text_msg_dsc,spec_tsb);
;  1151          1463  2 check (.status);
;  1152          1464  2
;  1153          1465  2 ! Now we look through the PUB chain, looking for PUBs that represent the
;  1154          1466  2 ! person who sent us the text.  The text is displayed in their viewport.
;  1155          1467  2
;  1156          1468  2 p = ..phn$gq_pubhead[0];
;  1157          1469  3 until .p eqla phn$gq_pubhead do (
;  1158          1470  3         if not .p[pub_v_temporary] and
;  1159          1471  3            not .p[pub_v_uhaveheld] and
```

```
; 1160          1472  3            not .p[pub_v_hasuheld]       then
; 1161          1473  3
; 1162          1474  3                 if phn$cmp_target(p[pub_b_tsb],spec_tsb) then
; 1163          1475  3                         phn$show_text(.p,text_dsc);
; 1164          1476  3
; 1165          1477  3            p = .p[pub_l_flink];
; 1166          1478  2 );
; 1167          1479  2
; 1168          1480  2 return;
; 1169          1481  2
; 1170          1482  1 end;


                                                0004 00000          .ENTRY  PHN$LISTEN, Save R2                             ; 1441
                                  5E      FF14  CE   9E 00002       MOVAB   -236(SP), SP                                    ;
                                  52        04  AC   D0 00007       MOVL    TEXT_MSG, R2                                    ; 1444
                04    B2                  62        00  3A 0000B       LOCC    #0, (R2), @4(R2)                                ; 1456
                                                    02  12 00010       BNEQ    1$                                              ;
                                                    51  D4 00012       CLRL    R1                                              ;
                                FC    AD          01  A1  9E 00014 1$:    MOVAB   1(R1), TEXT_DSC+4                               ;
                      50        04    A2          FC  AD  C3 00019       SUBL3   TEXT_DSC+4, 4(R2), R0                           ; 1457
                F8    AD                    50        62  A1 0001F       ADDW3   (R2), R0, TEXT_DSC                              ;
                      50        FC    AD          04  A2  C3 00024       SUBL3   4(R2), TEXT_DSC+4, R0                           ; 1458
                      62                    50        01  A3 0002A       SUBW3   #1, R0, (R2)                                    ;
                                                4004  8F  BB 0002E       PUSHR   #^M<R2,SP>                                      ; 1462
                                  0000G   CF            02  FB 00032       CALLS   #2, PHN$MAKE_TSB                                ;
                                          09            50  E8 00037       BLBS    STATUS, 2$                                      ; 1463
                                                        50  DD 0003A       PUSHL   STATUS                                          ;
                              00000000G   00            01  FB 0003C       CALLS   #1, LIB$SIGNAL                                  ;
                                          52    0000G   DF  D0 00043 2$:    MOVL    @PHN$GQ_PUBHEAD, P                              ; 1468
                                          50    0000G   CF  9E 00048 3$:    MOVAB   PHN$GQ_PUBHEAD, R0                              ; 1469
                                          50            52  D1 0004D       CMPL    P, R0                                           ;
                                                        2C  13 00050       BEQL    5$                                              ;
                                          50    00F0    C2  9E 00052       MOVAB   240(P), R0                                      ; 1470
                      1E                  60            C2  E0 00057       BBS     #2, (R0), 4$                                    ;
                                          1B            60  E8 0005B       BLBS    (R0), 4$                                        ; 1471
                      17                  60            01  E0 0005E       BBS     #1, (R0), 4$                                    ; 1472
                                                        5E  DD 00062       PUSHL   SP                                              ; 1474
                                                    0C  A2  9F 00064       PUSHAB  12(P)                                           ;
                                  0000G   CF            02  FB 00067       CALLS   #2, PHN$CMP_TARGET                              ;
                                          0A            50  E9 0006C       BLBC    R0, 4$                                          ;
                                                    F8  AD  9F 0006F       PUSHAB  TEXT_DSC                                        ; 1475
                                                        52  DD 00072       PUSHL   P                                               ;
                                  0000G   CF            02  FB 00074       CALLS   #2, PHN$SHOW_TEXT                               ;
                                          52            62  D0 00079 4$:    MOVL    (P), P                                          ; 1477
                                                        CA  11 0007C       BRB     3$                                              ; 1469
                                                            04 0007E 5$:    RET                                                     ; 1482

; Routine Size:  127 bytes,    Routine Base:  $CODE$ + 06BC
```

```
;  1172          1483  1 %sbttl 'PHN$HANGUP_CMD - Handle HANGUP Command'
;  1173          1484  1 !++
;  1174          1485  1 ! Functional Description:
;  1175          1486  1 !       This routine handles the HANGUP command, by which the user hangs
;  1176          1487  1 !       up the phone.  This cancels any and all conversations currently
;  1177          1488  1 !       in progress or being initiated.
;  1178          1489  1 !
;  1179          1490  1 ! Formal Parameters:
;  1180          1491  1 !       none
;  1181          1492  1 !
;  1182          1493  1 ! Implicit Inputs:
;  1183          1494  1 !       global data
;  1184          1495  1 !
;  1185          1496  1 ! Implicit Outputs:
;  1186          1497  1 !       global data
;  1187          1498  1 !
;  1188          1499  1 ! Returned Value:
;  1189          1500  1 !       none
;  1190          1501  1 !
;  1191          1502  1 ! Side Effects:
;  1192          1503  1 !
;  1193          1504  1 !--
;  1194          1505  1
;  1195          1506  1
;  1196          1507  2 global routine phn$hangup_cmd: novalue = begin
;  1197          1508  2
;  1198          1509  2 local
;  1199          1510  2         p: ref pub, p2: ref pub;
;  1200          1511  2
;  1201          1512  2
;  1202          1513  2 ! First we scan all the PUBs (but not our own) and break the link to
;  1203          1514  2 ! absolutely all of them.
;  1204          1515  2
;  1205          1516  2 p = ..phn$gq_pubhead[0];
;  1206          1517  3 until .p eqla phn$gq_pubhead do (
;  1207          1518  3         p2 = .p[pub_l_flink];
;  1208          1519  3         phn$break_link(.p,smb__hungup);
;  1209          1520  3         p = .p2;
;  1210          1521  2 );
;  1211          1522  2
;  1212          1523  2 ! Now we have to clear the calling/answering information in our PUB,
;  1213          1524  2 ! because we just cancelled any such activities.  Also refresh the screen.
;  1214          1525  2
;  1215          1526  2 p = .phn$gq_pubhead[0];
;  1216          1527  2 p[pub_v_calling] = false;
;  1217          1528  2 p[pub_v_answering] = false;
;  1218          1529  2 p[pub_l_busylink] = 0;
;  1219          1530  2 phn$fresh_screen(false);
;  1220          1531  2
;  1221          1532  2 return;
;  1222          1533  2
;  1223          1534  1 end;
```

```
                                    000C 00000          .ENTRY  PHN$HANGUP_CMD, Save R2,R3                      ; 1507
                          52    0000G DF   D0 00002     MOVL    @PHN$GQ_PUBHEAD, P                              ; 1516
                          50    0000G CF   9E 00007 1$: MOVAB   PHN$GQ_PUBHEAD, R0                             ; 1517
                          50          52   D1 0000C     CMPL    P, R0                                           ;
                                      11   13 0000F     BEQL    2$                                              ;
                          53          62   D0 00011     MOVL    (P), P2                                         ; 1518
                                      09   DD 00014     PUSHL   #9                                              ; 1519
                                      52   DD 00016     PUSHL   P                                               ;
                    0000G CF          02   FB 00018     CALLS   #2, PHN$BREAK_LINK                              ;
                          52          53   D0 0001D     MOVL    P2, P                                           ; 1520
                                      E5   11 00020     BRB     1$                                              ; 1517
                          52    0000G CF   D0 00022 2$: MOVL    PHN$GQ_PUBHEAD, P                               ; 1526
                    00F0  C2          18   8A 00027     BICB2   #24, 240(P)                                     ; 1528
                                00F8  C2   D4 0002C     CLRL    248(P)                                          ; 1529
                                      7E   D4 00030     CLRL    -(SP)                                           ; 1530
                    0000G CF          01   FB 00032     CALLS   #1, PHN$FRESH_SCREEN                            ;
                                           04 00037     RET                                                     ; 1534

; Routine Size:  56 bytes,    Routine Base:  $CODE$ + 073B
```

```
;  1225          1535  1 %sbttl 'PHN$HUNGUP - Person Hung Up on Us'
;  1226          1536  1 !++
;  1227          1537  1 ! Functional Description:
;  1228          1538  1 !       This steering message routine handles the hungup message,
;  1229          1539  1 !       which someone sends us when they hang up on us.  We break
;  1230          1540  1 !       any links we might have with that person.
;  1231          1541  1 !
;  1232          1542  1 ! Formal Parameters:
;  1233          1543  1 !       from_msg        Address of descriptor of complete node/user name
;  1234          1544  1 !                       spec of the person hanging up, followed by eofrom.
;  1235          1545  1 !
;  1236          1546  1 ! Implicit Inputs:
;  1237          1547  1 !       global data
;  1238          1548  1 !
;  1239          1549  1 ! Implicit Outputs:
;  1240          1550  1 !       global data
;  1241          1551  1 !
;  1242          1552  1 ! Returned Value:
;  1243          1553  1 !       none
;  1244          1554  1 !
;  1245          1555  1 ! Side Effects:
;  1246          1556  1 !
;  1247          1557  1 !--
;  1248          1558  1
;  1249          1559  1
;  1250          1560  2 global routine phn$hungup(from_msg): novalue = begin
;  1251          1561  2
;  1252          1562  2 bind
;  1253          1563  2         from_msg_dsc = .from_msg: descriptor;
;  1254          1564  2
;  1255          1565  2 local
;  1256          1566  2         status: long,
;  1257          1567  2         spec_tsb: tsb,
;  1258          1568  2         p: ref pub, p2: ref pub,
;  1259          1569  2         op: ref pub,                    ! Pointer to our PUB.
;  1260          1570  2         refresh: byte;
;  1261          1571  2
;  1262          1572  2
;  1263          1573  2 ! First we parse the sender's spec by making a TSB.
;  1264          1574  2
;  1265          1575  2 dec (from_msg_dsc[len]);
;  1266          1576  2 status = phn$make_tsb(from_msg_dsc,spec_tsb);
;  1267          1577  2 check (.status);
;  1268          1578  2
;  1269          1579  2 ! Now we loop through all of the PUBs (but not our own) and find ones
;  1270          1580  2 ! that represent the person who just hung up on us.
;  1271          1581  2
;  1272          1582  2 op = .phn$gq_pubhead[0];
;  1273          1583  2 p = ..phn$gq_pubhead[0];
;  1274          1584  2 refresh = false;
;  1275          1585  3 until .p eqla phn$gq_pubhead do (
;  1276          1586  3         p2 = .p[pub_l_flink];
;  1277          1587  3         if phn$cmp_target(p[pub_b_tsb],spec_tsb) then
;  1278          1588  4                 if .p eqla .op[pub_l_busylink] then (
;  1279          1589  4
;  1280          1590  4                         ! We are busy with this person.  If we're calling
;  1281          1591  4                         ! them, cancel the call.  If they're calling us,
```

```
;  1282          1592  4                           ! return the favor.
;  1283          1593  4
;  1284          1594  5                           if .op[pub_v_calling] then (
;  1285          1595  5                                   phn$break_call();
;  1286          1596  5                                   phn$inform(phn$_dead);
;  1287          1597  4                           );
;  1288          1598  5                           if .op[pub_v_answering] then (
;  1289          1599  5                                   op[pub_v_answering] = false;
;  1290          1600  5                                   op[pub_l_busylink] = 0;
;  1291          1601  5                                   phn$break_link(.p,smb__hungup);
;  1292          1602  5                                   phn$inform(phn$_cancal[);
;  1293          1603  4                           );
;  1294          1604  4                   ) else (
;  1295          1605  4
;  1296          1606  4                           ! We're already talking to this person.  Return
;  1297          1607  4                           ! the favor, and remember we have to refresh screen.
;  1298          1608  4
;  1299          1609  4                           phn$break_link(.p,smb__hungup);
;  1300          1610  4                           refresh = true;
;  1301          1611  3                   );
;  1302          1612  3
;  1303          1613  3           p = .p2;
;  1304          1614  2 );
;  1305          1615  2
;  1306          1616  2 ! If we broke any links with this person that actually represented a
;  1307          1617  2 ! conversation, we need to refresh the screen and tell the user.
;  1308          1618  2
;  1309          1619  3 if .refresh then (
;  1310          1620  3           phn$fresh_screen(false);
;  1311          1621  3           phn$inform(phn$_hungup,spec_tsb[tsb_q_tkndsc,
;  1312          1622  3                                           .spec_tsb[tsb_w_tkncount]]);
;  1313          1623  2 );
;  1314          1624  2
;  1315          1625  2 return;
;  1316          1626  2
;  1317          1627  1 end;
```

```
                                    00FC 00000           .ENTRY  PHN$HUNGUP, Save R2,R3,R4,R5,R6,R7          ; 1560
                              57    0000G CF   9E 00002  MOVAB   PHN$INFORM, R7                              ;
                              5E    FF1C  CE   9E 00007  MOVAB   -228(SP), SP                                ;
                                      04  BC   B7 0000C  DECW    @FROM_MSG                                   ; 1575
                                          5E   DD 0000F  PUSHL   SP                                          ; 1576
                                      04  AC   DD 00011  PUSHL   FROM_MSG                                   ;
                          0000G   CF      02   FB 00014  CALLS   #2, PHN$MAKE_TSB                            ;
                                  09          50   E8 00019  BLBS    STATUS, 1$                              ; 1577
                                              50   DD 0001C  PUSHL   STATUS                                  ;
                      00000000G   00          01   FB 0001E  CALLS   #1, LIB$SIGNAL                         ;
                              52    0000G CF   D0 00025 1$:   MOVL    PHN$GQ_PUBHEAD, OP                      ; 1582
                              54    0000G DF   D0 0002A       MOVL    @PHN$GQ_PUBHEAD, P                      ; 1583
                                          55   94 0002F       CLRB    REFRESH                                 ; 1584
                              50    0000G CF   9E 00031 2$:   MOVAB   PHN$GQ_PUBHEAD, R0                      ; 1585
                              50          54   D1 00036       CMPL    P, R0                                   ;
                                          5E   13 00039       BEQL    6$                                      ;
```

```
                         56              64  D0 0003B          MOVL    (P), P2                         ; 1586
                                         5E  DD 0003E          PUSHL   SP                              ; 1587
                                      0C A4  9F 00040          PUSHAB  12(P)                           ;
                   0000G  CF             02  FB 00043          CALLS   #2, PHN$CMP_TARGET              ;
                         49              50  E9 00048          BLBC    R0, 5$                          ;
                   00F8  C2              54  D1 0004B          CMPL    P, 248(OP)                      ; 1588
                                         36  12 00050          BNEQ    4$                              ;
                         53        00F0  C2  9E 00052          MOVAB   240(OP), R3                     ; 1594
                      0E 63              03  E1 00057          BBC     #3, (R3), 3$                    ;
                   0000G  CF             00  FB 0005B          CALLS   #0, PHN$BREAK_CALL              ; 1595
                             00000000G   8F  DD 00060          PUSHL   #PHN$_DEAD                      ; 1596
                         67              01  FB 00066          CALLS   #1, PHN$INFORM                  ;
                      27 63              04  E1 00069 3$:      BBC     #4, (R3), 5$                    ; 1598
                         63              10  8A 0006D          BICB2   #16, (R3)                       ; 1599
                                   00F8  C2  D4 00070          CLRL    248(OP)                         ; 1600
                                         09  DD 00074          PUSHL   #9                              ; 1601
                                         54  DD 00076          PUSHL   P                               ;
                   0000G  CF             02  FB 00078          CALLS   #2, PHN$BREAK_LINK              ;
                             00000000G   8F  DD 0007D          PUSHL   #PHN$_CANCALL                   ; 1602
                         67              01  FB 00083          CALLS   #1, PHN$INFORM                  ;
                                         0C  11 00086          BRB     5$                              ; 1588
                                         09  DD 00088 4$:      PUSHL   #9                              ; 1609
                                         54  DD 0008A          PUSHL   P                               ;
                   0000G  CF             02  FB 0008C          CALLS   #2, PHN$BREAK_LINK              ;
                         55              01  90 00091          MOVB    #1, REFRESH                     ; 1610
                         54              56  D0 00094 5$:      MOVL    P2, P                           ; 1613
                                         98  11 00097          BRB     2$                              ; 1585
                         18              55  E9 00099 6$:      BLBC    REFRESH, 7$                     ; 1619
                                         7E  D4 0009C          CLRL    -(SP)                           ; 1620
                   0000G  CF             01  FB 0009E          CALLS   #1, PHN$FRESH_SCREEN            ;
                         50        02    AE  3C 000A3          MOVZWL  SPEC_TSB+2, R0                  ; 1622
                                   04 AE40   7F 000A7          PUSHAQ  SPEC_TSB+4[R0]                  ;
                             00000000G   8F  DD 000AB          PUSHL   #PHN$_HUNGUP                    ;
                         67              02  FB 000B1          CALLS   #2, PHN$INFORM                  ;
                                         04 000B4 7$:          RET                                     ; 1627

; Routine Size:  181 bytes,    Routine Base:  $CODE$ + 0773

; 1318          1628  1
; 1319          1629  0 end eludom


                                                               .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $PLIT$                        113  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                       2088  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                         307  NOVEC,  WRT,  RD ,NOEXE,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        19         0       581         00:00.7
```

```
;                                    COMMAND QUALIFIERS

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

; Size:         2088 code + 420 data bytes
; Run Time:       00:26.1
; Elapsed Time:   01:41.2
; Lines/CPU Min:   3739
; Lexemes/CPU-Min: 28202
; Memory Used:  154 pages
; Compilation Complete
```

0304 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PHONE

PATSYM
LIS

PHONE
MAP

BASICCMDS
LIS

PATSTO
LIS

PATVEC
LIS

PHONEREQ
REQ

PATWRT
LIS

FILECMDS
LIS