```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB   AAA         AAA  SSS            RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA  SSS            RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA  SSS            RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA  SSS            RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA  SSS            RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA  SSS            RRR         RRR         TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS    RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS    RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS    RRRRRRRRRRR             TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA              SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA              SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA              SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA              SSS  RRR      RRR            TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA              SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA              SSS  RRR      RRR            TTT        LLL
BBB         BBB   AAA         AAA              SSS  RRR      RRR            TTT        LLL
BBBBBBBBBBBB      AAA         AAA  SSSSSSSSSSS     RRR         RRR         TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA  SSSSSSSSSSS     RRR         RRR         TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA  SSSSSSSSSSS     RRR         RRR         TTT        LLLLLLLLLLLLLLL
```

FILEID**BASSYS

```
BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS  YY      YY    SSSSSSSS
BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS  YY      YY    SSSSSSSS
BB      BB   AA      AA  SS          SS          YY      YY   SS
BB      BB   AA      AA  SS          SS          YY      YY   SS
BB      BB   AA      AA  SS          SS            YY  YY     SS
BB      BB   AA      AA  SS          SS            YY  YY     SS
BBBBBBBB     AA      AA    SSSSSS      SSSSSS        YY         SSSSSS
BBBBBBBB     AA      AA    SSSSSS      SSSSSS        YY         SSSSSS
BB      BB   AAAAAAAAAA          SS          SS      YY               SS
BB      BB   AAAAAAAAAA          SS          SS      YY               SS
BB      BB   AA      AA          SS          SS      YY               SS      ....
BB      BB   AA      AA          SS          SS      YY               SS      ....
BBBBBBBB     AA      AA  SSSSSSSS    SSSSSSSS        YY      SSSSSSSS        ....
BBBBBBBB     AA      AA  SSSSSSSS    SSSSSSSS        YY      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 MODULE BAS$SYS (                                    ! RSTS SYS Function
;     2          0002  0                  IDENT = '1-014'                    ! File: BASSYS.B32 Edit: PL1014
;     3          0003  0                  ) =
;     4          0004  1 BEGIN
;     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-11 BASIC RSTS COMPATABILITY
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains the RSTS-compatable SYS function.
;    36          0036  1 !       Some of the more complex functions do calls to BPA routines.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  VAX-11 User Mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: 01-OCT-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.
;    45          0045  1 ! 1-002 - Define BPA$A_HISEG as 0, to disable the checking for the end
;    46          0046  1 !           of the compatability-mode high segment, and allocate
;    47          0047  1 !           some global cells for ASSIGN/DEASSIGN.  JBS 02-OCT-1979
;    48          0048  1 ! 1-003 - Instead of passing the XRB to the message send/receive code,
;    49          0049  1 !           pass the relevant fields of the XRB.  Also, BPA$_HISEG is
;    50          0050  1 !           no longer needed.  JBS 04-OCT-1979
;    51          0051  1 ! 1-004 - Debug message send/receive.  JBS 05-OCT-1979
;    52          0052  1 ! 1-005 - Continue debugging message send/receive.  JBS 07-OCT-1979
;    53          0053  1 ! 1-006 - Fix some errors in calling terminal functions.  JBS 12-OCT-1979
;    54          0054  1 ! 1-007 - Handle short parameter strings correctly.  JBS 17-OCT-1979
;    55          0055  1 ! 1-008 - Add core common.  JBS 03-DEC-1979
;    56          0056  1 ! 1-009 - Add FSS.  JBS 04-DEC-1979
;    57          0057  1 ! 1-010 - Add setting priority.  JBS 04-DEC-1979
```

```
;    58          0058  1 ! 1-011 - Make "exit with no prompt" exit more quietly, fix a typo in putting
;    59          0059  1 !           into core common, and make ODT submode give an error message, since
;    60          0060  1 !           there isn't time to make it work right for this release.  JBS 26-FEB-1980
;    61          0061  1 ! 1-012 - Put in small send and receive. All four functions. FM 24-FEB-81.
;    62          0062  1 ! 1-013 - LIB$STOP should be declared EXTERNAL.  PL 20-Nov-81
;    63          0063  1 ! 1-014 - Call BAS$$STOP to signal errors instead of BAS$$STOP_IO.  PLL 16-Jun-1982
;    64          0064  1 !--
;    65          0065  1
;    66          0066  1 !<BLF/PAGE>
```

```
;    68          0067  1 !
;    69          0068  1 ! SWITCHES:
;    70          0069  1 !
;    71          0070  1
;    72          0071  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    73          0072  1
;    74          0073  1 !
;    75          0074  1 ! LINKAGES:
;    76          0075  1 !
;    77          0076  1
;    78          0077  1 REQUIRE 'RTLIN:OTSLNK';                         ! I/O Linkages
;    79          0506  1
;    80          0507  1 !
;    81          0508  1 ! TABLE OF CONTENTS:
;    82          0509  1 !
;    83          0510  1
;    84          0511  1 FORWARD ROUTINE
;    85          0512  1     BAS$SYS : NOVALUE,                          ! Do a SYS() function
;    86          0513  1     BAS$$UUO : NOVALUE;                         ! Do a UUO sub-function
;    87          0514  1
;    88          0515  1 !
;    89          0516  1 ! INCLUDE FILES:
;    90          0517  1 !
;    91          0518  1
;    92          0519  1 REQUIRE 'RTLML:OTSLUB';                         ! LUB definitions
;    93          0659  1
;    94          0660  1 REQUIRE 'RTLIN:BPASTRUCT';                      ! Structureing definitions
;    95          0751  1
;    96          0752  1 REQUIRE 'RTLIN:BPAFQBDEF';                      ! FIRQB definitions
;    97          0876  1
;    98          0877  1 REQUIRE 'RTLIN:BPAFSBDEF';                      ! FSB definitions
;    99          1008  1
;   100          1009  1 REQUIRE 'RTLIN:BPAFUNDEF';                      ! Function definitions
;   101          1259  1
;   102          1260  1 REQUIRE 'RTLIN:RTLPSECT';                       ! Macros for defining psects
;   103          1355  1
;   104          1356  1 LIBRARY 'RTLSTARLE';                            ! System symbols
;   105          1357  1
;   106          1358  1 !
;   107          1359  1 ! MACROS:
;   108          1360  1 !
;   109          1361  1 !       NONE
;   110          1362  1 !
;   111          1363  1 ! EQUATED SYMBOLS:
;   112          1364  1 !
;   113          1365  1 !       NONE
;   114          1366  1 !
;   115          1367  1 ! PSECTS:
;   116          1368  1 !
;   117          1369  1 DECLARE_PSECTS (BAS);                           ! Declare psects for BAS$ facility
;   118          1370  1 !
;   119          1371  1 ! OWN STORAGE:
;   120          1372  1 !
;   121          1373  1 !       NONE
;   122          1374  1 !
;   123          1375  1 ! EXTERNAL REFERENCES:
;   124          1376  1 !
```

```
;  125          1377  1
;  126          1378  1 EXTERNAL ROUTINE
;  127          1379  1     LIB$STOP : NOVALUE,                          ! Signal a fatal error
;  128          1380  1     STR$COPY_DX,                                 ! Copy a string by descriptor
;  129          1381  1     STR$COPY_R,                                  ! Copy a string by reference
;  130          1382  1     STR$FREE1_DX,                                ! Free a string
;  131          1383  1     STR$CONCAT,                                  ! Concatenate strings
;  132          1384  1     LIB$GET_COMMON,                              ! Fetch from process common
;  133          1385  1     LIB$PUT_COMMON,                              ! Store in process common
;  134          1386  1     BAS$$STOP : NOVALUE,                         ! Signal fatal error
;  135          1387  1     BAS$RCTRLO : NOVALUE,                        ! Clear control O
;  136          1388  1     BAS$NOECHO : NOVALUE,                        ! Turn off echoing
;  137          1389  1     BAS$ONECHR : NOVALUE,                        ! Enter ODT submode
;  138          1390  1     BAS$CANTYPAHEAD : NOVALUE,                   ! Cancel type-ahead
;  139          1391  1     BAS$CTRLC : NOVALUE,                         ! Enable control-C trapping
;  140          1392  1     BAS$ECHO : NOVALUE,                          ! Turn of echoing
;  141          1393  1     BAS$ERT : NOVALUE,                           ! Return error message
;  142          1394  1     BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,          ! Load register CCB
;  143          1395  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,            ! Done with register CCB
;  144          1396  1     BAS$$STOP_IO : NOVALUE,                      ! Signal fatal I/O error
;  145          1397  1     BPA$MESAG,                                   ! Do send/receive
;  146          1398  1     BPA$ASSIGN,                                  ! Assign a device
;  147          1399  1     BPA$DEASSIGN,                                ! Deassign a device
;  148          1400  1     BPA$DEASS_ALL,                               ! Deassign all devices
;  149          1401  1     BPA$SET_DEF,                                 ! Set default
;  150          1402  1     BPA$FSS,                                     ! File string scan
;  151          1403  1     BPA$FREE_BLOCK,                              ! Free heap storage
;  152          1404  1     BPA$SET_PRI;                                 ! Change priority
;  153          1405  1
;  154          1406  1 !+
;  155          1407  1 ! The following are the error codes used in this module.
;  156          1408  1 !-
;  157          1409  1
;  158          1410  1 EXTERNAL LITERAL
;  159          1411  1     BAS$K_ILLIO_CHA : UNSIGNED (8),              ! Illegal I/O channel
;  160          1412  1     BAS$K_IO_CHANOT : UNSIGNED (8),              ! I/O channel not open
;  161          1413  1     BAS$K_ILLSYSUSA : UNSIGNED (8),              ! Illegal SYS$() usage
;  162          1414  1     BAS$K_ILLBYTCOU : UNSIGNED (8),              ! Illegal byte count for I/O
;  163          1415  1     BAS$K_NO_ROOUSE : UNSIGNED (8),              ! No room fcr user on device
;  164          1416  1     BAS$K_MISSPEFEA : UNSIGNED (8);              ! Missing special feature
;  165          1417  1
```

```
;  167          1418  1 GLOBAL ROUTINE BAS$SYS (                      ! Do a SYS() function
;  168          1419  1         RESULT_STR,                           ! Result of the SYS() function
;  169          1420  1         CODE_STR                              ! String that tells what to do
;  170          1421  1     ) : NOVALUE =
;  171          1422  1
;  172          1423  1 !++
;  173          1424  1 ! FUNCTIONAL DESCRIPTION:
;  174          1425  1 !
;  175          1426  1 !       Do a RSTS/E compatable SYS() function.
;  176          1427  1 !
;  177          1428  1 ! FORMAL PARAMETERS:
;  178          1429  1 !
;  179          1430  1 !       RESULT_STR.wz.dx        Result of the SYS() function
;  180          1431  1 !       CODE_STR.rz.dx          A string of bytes which specify what to do.
;  181          1432  1 !
;  182          1433  1 ! IMPLICIT INPUTS:
;  183          1434  1 !
;  184          1435  1 !       NONE
;  185          1436  1 !
;  186          1437  1 ! IMPLICIT OUTPUTS:
;  187          1438  1 !
;  188          1439  1 !       NONE
;  189          1440  1 !
;  190          1441  1 ! ROUTINE VALUE:
;  191          1442  1 !
;  192          1443  1 !       Depends on the function, see above.  Where the result is not
;  193          1444  1 !       defined the null string is returned.
;  194          1445  1 !
;  195          1446  1 ! SIDE EFFECTS:
;  196          1447  1 !
;  197          1448  1 !       Signals if an error is encountered.
;  198          1449  1 !
;  199          1450  1 !--
;  200          1451  1
;  201          1452  2     BEGIN
;  202          1453  2
;  203          1454  2     MAP
;  204          1455  2         CODE_STR : REF BLOCK [8, BYTE];
;  205          1456  2
;  206          1457  2     BIND
;  207          1458  2         FIRST_BYTE = CODE_STR [DSC$A_POINTER] : REF VECTOR [, BYTE],
;  208          1459  2         STR_LENGTH = .CODE_STR [DSC$W_LENGTH];
;  209          1460  2
;  210          1461  2     LOCAL
;  211          1462  2         RET_STRING : BLOCK [8, BYTE];
;  212          1463  2
;  213          1464  2     RET_STRING [DSC$W_LENGTH] = 0;
;  214          1465  2     RET_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_Z;
;  215          1466  2     RET_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
;  216          1467  2     RET_STRING [DSC$A_POINTER] = 0;
;  217          1468  2 !+
;  218          1469  2 ! Dispatch on the first byte of the string.
;  219          1470  2 !-
;  220          1471  2
;  221          1472  3     IF (STR_LENGTH EQLU 0)
;  222          1473  2     THEN
;  223          1474  2         BAS$$STOP (BAS$K_ILLSYSUSA)
```

```
;  224          1475  2      ELSE
;  225          1476  2
;  226          1477  2          CASE .FIRST_BYTE [0] FROM 0 TO 12 OF
;  227          1478  2              SET
;  228          1479  2
;  229          1480  2              [0] :                                   ! Cancel control O
;  230          1481  2                  BAS$RCTRLO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
;  231          1482  2
;  232          1483  2              [1] :                                   ! Enter TAPE mode
;  233          1484  2                  BAS$$STOP (BAS$K_MISSPEFEA);
;  234          1485  2
;  235          1486  2              [2] :                                   ! Enable echoing
;  236          1487  2                  BAS$ECHO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
;  237          1488  2
;  238          1489  2              [3] :                                   ! Disable echoing
;  239          1490  2                  BAS$NOECHO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
;  240          1491  2
;  241          1492  2              [4] :                                   ! ODT submode
;  242          1493  3                  BEGIN
;  243          1494  3                  BAS$$STOP (BAS$K_MISSPEFEA);
;  244          1495  3                  BAS$ONECHR ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
;  245          1496  2                  END;
;  246          1497  2
;  247          1498  2              [5] :                                   ! Exit with no prompt
;  248          1499  2                  $EXIT (CODE = SS$_NORMAL);
;  249          1500  2
;  250          1501  2              [6] :                                   ! Call file processor
;  251          1502  2                  BAS$$UUO (RET_STRING, .CODE_STR);
;  252          1503  2
;  253          1504  2              [7] :                                   ! Get core common
;  254          1505  3                  BEGIN
;  255          1506  3
;  256          1507  3                  LOCAL
;  257          1508  3                      STATUS;
;  258          1509  3
;  259          1510  3                  STATUS = LIB$GET_COMMON (RET_STRING);
;  260          1511  3
;  261          1512  3                  IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
;  262          1513  3
;  263          1514  2                  END;
;  264          1515  2
;  265          1516  2              [8] :                                   ! Put core common
;  266          1517  3                  BEGIN
;  267          1518  3
;  268          1519  3                  LOCAL
;  269          1520  3                      STATUS,
;  270          1521  3                      LEN,
;  271          1522  3                      DESC : BLOCK [8, BYTE];
;  272          1523  3
;  273          1524  3                  LEN = STR_LENGTH - 1;
;  274          1525  3
;  275          1526  3                  IF (.LEN GTR 127) THEN LEN = 0;
;  276          1527  3
;  277          1528  3                  DESC [DSC$W_LENGTH] = .LEN;
;  278          1529  3                  DESC [DSC$B_DTYPE] = DSC$K_DTYPE_Z;
;  279          1530  3                  DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  280          1531  3                  DESC [DSC$A_POINTER] = FIRST_BYTE [1];
```

```
;  281          1532  3              STATUS = LIB$PUT_COMMON (DESC);
;  282          1533  3
;  283          1534  3              IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
;  284          1535  3
;  285          1536  2              END;
;  286          1537  2
;  287          1538  2          [9] :                                   ! Exit and clear program
;  288          1539  2              $EXIT (CODE = SS$_NORMAL);
;  289          1540  2
;  290          1541  2          [10] :                                  ! Special implementation
;  291          1542  2              BAS$$STOP (BAS$K_MISSPEFEA);
;  292          1543  2
;  293          1544  2          [11] :                                  ! Cancel type ahead
;  294          1545  2              BAS$CANTYPAHEAD ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
;  295          1546  2
;  296          1547  2          [12] :                                  ! Info on last opened file
;  297          1548  2              BAS$$STOP (BAS$K_MISSPEFEA);
;  298          1549  2
;  299          1550  2          [OUTRANGE] :                            ! Not defined
;  300          1551  2              BAS$$STOP (BAS$K_ILLSYSUSA);
;  301          1552  2          TES;
;  302          1553  2
;  303          1554  2      STR$COPY_DX (.RESULT_STR, RET_STRING);
;  304          1555  2      STR$FREE1_DX (RET_STRING);
;  305          1556  2      RETURN;
;  306          1557  1      END;                                        ! of routine BAS$SYS

                                                .TITLE  BAS$SYS
                                                .IDENT  \1-014\

                                                .EXTRN  LIB$STOP, STR$COPY_DX
                                                .EXTRN  STR$COPY_R, STR$FREE1_DX
                                                .EXTRN  STR$CONCAT, LIB$GET_COMMON
                                                .EXTRN  LIB$PUT_COMMON, BAS$$STOP
                                                .EXTRN  BAS$RCTRLO, BAS$NOECHO
                                                .EXTRN  BAS$ONECHR, BAS$CANTYPAHEAD
                                                .EXTRN  BAS$CTRLC, BAS$ECHO
                                                .EXTRN  BAS$ERT, BAS$$CB_PUSH
                                                .EXTRN  BAS$$CB_POP, BAS$$STOP_IO
                                                .EXTRN  BPA$MESAG, BPA$ASSIGN
                                                .EXTRN  BPA$DEASSIGN, BPA$DEASS_ALL
                                                .EXTRN  BPA$SET_DEF, BPA$FSS
                                                .EXTRN  BPA$FREE_BLOCK, BPA$SET_PRI
                                                .EXTRN  BAS$K_ILLIO_CHA
                                                .EXTRN  BAS$K_IO_CHANOT
                                                .EXTRN  BAS$K_ILLSYSUSA
                                                .EXTRN  BAS$K_ILLBYTCOU
                                                .EXTRN  BAS$K_NO_ROOUSE
                                                .EXTRN  BAS$K_MISSPEFEA
                                                .EXTRN  SYS$EXIT

                                                .PSECT  _BAS$CODE,NOWRT,  SHR,  PIC,2

                        001C 00000              .ENTRY  BAS$SYS, Save R2,R3,R4                  ; 1418
54 00000000G    00   9E 00002              MOVAB   BAS$$STOP, R4                           ;
5E              10   C2 00009              SUBL2   #16, SP                                 ;
```

```
                    50      08   AC              04   C1 0000C          ADDL3   #4, CODE_STR, R0                        ;  1458
                                 53          08  BC   3C 00011          MOVZWL  @CODE_STR, R3                           ;  1459
                            08   AE 02000000    8F   D0 00015          MOVL    #33554432, RET_STRING                   ;  1464
                                              0C AE   D4 0001D          CLRL    RET_STRING+4                            ;  1467
                                                 53   D5 00020          TSTL    R3                                      ;  1472
                                                 21   13 00022          BEQL    2$                                      ;
                                      52         60   D0 00024          MOVL    (R0), R2                                ;  1477
                              0C      00         62   8F 00027          CASEB   (R2), #0, #12                           ;
                    004D    0037          00EE      0021      0002B 1$:  .WORD   3$-1$,-                                 ;
                    008D    0080          00CD      0063      00033              27$-1$,-                                ;
                    00D8    00EE          00CD      0099      0003B              6$-1$,-                                 ;
                                                    00EE      00043              9$-1$,-                                 ;
                                                                                 12$-1$,-                                ;
                                                                                 22$-1$,-                                ;
                                                                                 15$-1$,-                                ;
                                                                                 17$-1$,-                                ;
                                                                                 18$-1$,-                                ;
                                                                                 22$-1$,-                                ;
                                                                                 27$-1$,-                                ;
                                                                                 24$-1$,-                                ;
                                                                                 27$-1$                                  ;
                                      7E        00G  8F   9A 00045 2$:  MOVZBL  #BAS$K_ILLSYSUSA, -(SP)                 ;  1551
                                                00D1 31   00049        BRW     28$                                     ;
                                      02             53   B1 0004C 3$:  CMPW    R3, #2                                  ;  1481
                                                     06   1F 0004F        BLSSU   4$                                      ;
                                      7E        01   A2   9A 00051        MOVZBL  1(R2), -(SP)                            ;
                                                     02   11 00055        BRB     5$                                      ;
                                                     7E   D4 00057 4$:    CLRL    -(SP)                                   ;
                          00000000G   00             01   FB 00059 5$:    CALLS   #1, BAS$RCTRLO                          ;
                                                     54   11 00060        BRB     16$                                     ;
                                      02             53   B1 00062 6$:    CMPW    R3, #2                                  ;  1487
                                                     06   1F 00065        BLSSU   7$                                      ;
                                      7E        01   A2   9A 00067        MOVZBL  1(R2), -(SP)                            ;
                                                     02   11 0006B        BRB     8$                                      ;
                                                     7E   D4 0006D 7$:    CLRL    -(SP)                                   ;
                          00000000G   00             01   FB 0006F 8$:    CALLS   #1, BAS$ECHO                            ;
                                                     7E   11 00076        BRB     21$                                     ;
                                      02             53   B1 00078 9$:    CMPW    R3, #2                                  ;  1490
                                                     06   1F 0007B        BLSSU   10$                                     ;
                                      7E        01   A2   9A 0007D        MOVZBL  1(R2), -(SP)                            ;
                                                     02   11 00081        BRB     11$                                     ;
                                                     7E   D4 00083 10$:   CLRL    -(SP)                                   ;
                          00000000G   00             01   FB 00085 11$:   CALLS   #1, BAS$NOECHO                         ;
                                                     73   11 0008C        BRB     23$                                     ;
                                      7E        00G  8F   9A 0008E 12$:   MOVZBL  #BAS$K_MISSPEFEA, -(SP)                 ;  1494
                                      64             01   FB 00092        CALLS   #1, BA5$$STOP                           ;
                                      02             53   B1 00095        CMPW    R3, #2                                  ;  1495
                                                     06   1F 00098        BLSSU   13$                                     ;
                                      7E        01   A2   9A 0009A        MOVZBL  1(R2), -(SP)                            ;
                                                     02   11 0009E        BRB     14$                                     ;
                                                     7E   D4 000A0 13$:   CLRL    -(SP)                                   ;
                          00000000G   00             01   FB 000A2 14$:   CALLS   #1, BAS$ONECHR                         ;
                                                     75   11 000A9        BRB     29$                                     ;  1477
                                                08   AC   DD 000AB 15$:   PUSHL   CODE_STR                               ;  1502
                                                0C   AE   9F 000AE        PUSHAB  RET_STRING                              ;
                                0000V           CF   02   FB 000B1        CALLS   #2, BAS$$UUO                            ;
                                                     68   11 000B6 16$:   BRB     29$                                     ;
```

```
                        08   AE  9F 000B8 17$:    PUSHAB  RET_STRING                          ; 1510
00000000G   00              01  FB 000BB         CALLS   #1, LIB$GET_COMMON                  ;
                            26  11 000C2         BRB     20$                                 ; 1512
                    50  FF  A3  9E 000C4 18$:    MOVAB   -1(R3), LEN                         ; 1524
0000007F    8F              50  D1 000C8         CMPL    LEN, #127                           ; 1526
                            02  15 000CF         BLEQ    19$                                 ;
                            50  D4 000D1         CLRL    LEN                                 ;
                6E          50  B0 000D3 19$:    MOVW    LEN, DESC                           ; 1528
          02    AE    0100  8F  B0 000D6         MOVW    #256, DESC+2                        ; 1529
          04    AE      01  A2  9E 000DC         MOVAB   1(R2), DESC+4                       ; 1531
                        5E  DD 000E1             PUSHL   SP                                  ; 1532
00000000G   00              01  FB 000E3         CALLS   #1, LIB$PUT_COMMON                  ;
                    33      50  E8 000EA 20$:    BLBS    STATUS, 29$                         ; 1534
                            50  DD 000ED         PUSHL   STATUS                              ;
00000000G   00              01  FB 000EF         CALLS   #1, LIB$STOP                        ;
                            28  11 000F6 21$:    BRB     29$                                 ; 1477
                            01  DD 000F8 22$:    PUSHL   #1                                  ; 1539
00000000G   00              01  FB 000FA         CALLS   #1, SYS$EXIT                        ;
                            1D  11 00101 23$:    BRB     29$                                 ;
                    02      53  B1 00103 24$:    CMPW    R3, #2                              ; 1545
                            06  1F 00106         BLSSU   25$                                 ;
                7E      01  A2  9A 00108         MOVZBL  1(R2), -(SP)                        ;
                            02  11 0010C         BRB     26$                                 ;
                            7E  D4 0010E 25$:    CLRL    -(SP)                               ;
00000000G   00              01  FB 00110 26$:    CALLS   #1, BAS$CANTYPAHEAD                 ;
                            07  11 00117         BRB     29$                                 ;
                7E      00G 8F  9A 00119 27$:    MOVZBL  #BAS$K_MISSPEFEA, -(SP)             ; 1548
                64          01  FB 0011D 28$:    CALLS   #1, BAS$$STOP                       ;
                        08  AE  9F 00120 29$:    PUSHAB  RET_STRING                          ; 1554
                        04  AC  DD 00123         PUSHL   RESULT_STR                          ;
00000000G   00              02  FB 00126         CALLS   #2, STR$COPY_DX                     ;
                        08  AE  9F 0012D         PUSHAB  RET_STRING                          ; 1555
00000000G   00              01  FB 00130         CALLS   #1, STR$FREE1_DX                    ;
                                04 00137         RET                                         ; 1557

; Routine Size:  312 bytes,    Routine Base:  _BAS$CODE + 0000

;  307          1558  1
```

```
;  309          1559  1 ROUTINE BAS$$UUO (                                    ! Do a SYS(CHR$(6)) function
;  310          1560  1         RESULT_STR,                                   ! Result of the SYS() call
;  311          1561  1         CODE_STR                                      ! String that tells what to do
;  312          1562  1     ) : NOVALUE =
;  313          1563  1
;  314          1564  1 !++
;  315          1565  1 ! FUNCTIONAL DESCRIPTION:
;  316          1566  1 !
;  317          1567  1 !       Do a RSTS/E compatable SYS() function, where the first byte of
;  318          1568  1 !       the string is a 6.
;  319          1569  1 !
;  320          1570  1 ! FORMAL PARAMETERS:
;  321          1571  1 !
;  322          1572  1 !       RESULT_STR.wz.dx        The result of the SYS() function
;  323          1573  1 !       CODE_STR.rz.dx          A string of bytes which specify what to do.
;  324          1574  1 !
;  325          1575  1 ! IMPLICIT INPUTS:
;  326          1576  1 !
;  327          1577  1 !       NONE
;  328          1578  1 !
;  329          1579  1 ! IMPLICIT OUTPUTS:
;  330          1580  1 !
;  331          1581  1 !       NONE
;  332          1582  1 !
;  333          1583  1 ! ROUTINE VALUE:
;  334          1584  1 !
;  335          1585  1 !       Depends on the function, see above.  Where the result is not
;  336          1586  1 !       specified, the user's string is set to null.
;  337          1587  1 !
;  338          1588  1 ! SIDE EFFECTS:
;  339          1589  1 !
;  340          1590  1 !       Signals if an error is encountered.
;  341          1591  1 !
;  342          1592  1 !--
;  343          1593  1
;  344          1594  2     BEGIN
;  345          1595  2
;  346          1596  2     EXTERNAL ROUTINE
;  347          1597  2         R50ASC : NOVALUE;
;  348          1598  2
;  349          1599  2     MAP
;  350          1600  2         CODE_STR : REF BLOCK [8, BYTE];
;  351          1601  2
;  352          1602  2     LOCAL
;  353          1603  2         RET_STRING : BLOCK [8, BYTE],
;  354          1604  2         FIRQB : BLOCK [512 + 41 + 2, BYTE] FIELD (FQB$FIELDS),
;  355          1605  2         FIRQB_DESC : BLOCK [8, BYTE];
;  356          1606  2
;  357          1607  2     FIRQB_DESC [DSC$W_LENGTH] = 512 + 41;
;  358          1608  2     FIRQB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_BU;
;  359          1609  2     FIRQB_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  360          1610  2     FIRQB_DESC [DSC$A_POINTER] = FIRQB [2, B_];
;  361          1611  2     CH$COPY (.CODE_STR [DSC$W_LENGTH], .CODE_STR [DSC$A_POINTER], 0, 512 + 41, FIRQB [2, B_]);
;  362          1612  2     RET_STRING [DSC$W_LENGTH] = 0;
;  363          1613  2     RET_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_Z;
;  364          1614  2     RET_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
;  365          1615  2     RET_STRING [DSC$A_POINTER] = 0;
```

```
;  366          1616  2 !+
;  367          1617  2 ! Dispatch on the second byte of the string.
;  368          1618  2 !-
;  369          1619  2
;  370          1620  2     CASE .FIRQB [FQB$B_FUNCTION] FROM FUN$K_MINUUO TO FUN$K_MAXUUO OF
;  371          1621  2         SET
;  372          1622  2
;  373          1623  2         [FUN$K_UUCCT] :                         ! Control C trap enable
;  374          1624  2             BAS$CTRLC ();
;  375          1625  2
;  376          1626  2         [FUN$K_UUERR] :                         ! Get error message
;  377          1627  3             BEGIN
;  378          1628  3
;  379          1629  3             LOCAL
;  380          1630  3                 ERR_STRING : BLOCK [8, BYTE],
;  381          1631  3                 HEADER : BLOCK [8, BYTE];
;  382          1632  3
;  383          1633  3             ERR_STRING [DSC$W_LENGTH] = 0;
;  384          1634  3             ERR_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  385          1635  3             ERR_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
;  386          1636  3             ERR_STRING [DSC$A_POINTER] = 0;
;  387          1637  3             BAS$ERT (ERR_STRING, .FIRQB [FQB$B_ERRNUM]);
;  388          1638  3             HEADER [DSC$W_LENGTH] = 2;
;  389          1639  3             HEADER [DSC$B_DTYPE] = DSC$K_DTYPE_BU;
;  390          1640  3             HEADER [DSC$B_CLASS] = DSC$K_CLASS_S;
;  391          1641  3             HEADER [DSC$A_POINTER] = UPLIT (BYTE (0, 0));
;  392          1642  3             STR$CONCAT (RET_STRING, HEADER, ERR_STRING);
;  393          1643  3             STR$FREE1_DX (ERR_STRING);
;  394          1644  2             END;
;  395          1645  2
;  396          1646  2         [FUN$K_UUMES1] :                        ! Small message send/receive
;  397          1647  3             BEGIN
;  398          1648  3
;  399          1649  3             LITERAL
;  400          1650  3                 K_SML_SEND = -1,
;  401          1651  3                 K_SML_REMREC = 0,
;  402          1652  3                 K_SML_DCLREC_REC = 1,
;  403          1653  3                 K_SML_REC = 2;
;  404          1654  3
;  405          1655  3             LOCAL
;  406          1656  3                 BYTXFR,                             ! Number of bytes actually transferred
;  407          1657  3                 ASCII_LOGNAM : VECTOR [6, BYTE],        ! A buffer to put the translated R50 name.
;  408          1658  3                 RES_STRING : VECTOR [30, BYTE], ! Some place to put the result string temporarily.
;  409          1659  3                 MESAG : VECTOR [20, BYTE];      ! A temp. place to put the message.
;  410          1660  3
;  411          1661  3 !+
;  412          1662  3 ! Translate the name passed to ASCII.
;  413          1663  3 !-
;  414          1664  3             R50ASC (%REF (6), FIRQB [FQB$T_RCVNAM], ASCII_LOGNAM);
;  415          1665  3
;  416          1666  3             IF .ASCII_LOGNAM [0] EQL %C'?' THEN BAS$$STOP_IO (BAS$K_ILLSYSUSA);
;  417          1667  3
;  418          1668  3 !+
;  419          1669  3 ! Do each function seperately
;  420          1670  3 !-
;  421          1671  3
;  422          1672  3             CASE .FIRQB [FQB$B_SUBFUN] FROM K_SML_SEND TO K_SML_REC OF
```

```
;  423     1673  3            SET
;  424     1674  3
;  425     1675  3            [K_SML_SEND] :
;  426     1676  4                BEGIN
;  427     1677  4                CH$MOVE (20, FIRQB [10, B_], MESAG);
;  428     1678  4                CH$MOVE (20, MESAG, FIRQB [12, B_]);          !The message
;  429     1679  4                CH$MOVE (6, ASCII_LOGNAM, FIRQB [FQB$T_RCVNAM]);    !Logical name
;  430     1680  3                END;
;  431     1681  3
;  432     1682  3            [K_SML_REMREC] :
;  433     1683  3                1;                              ! Looks the same, so don't do anything.
;  434     1684  3
;  435     1685  3            [K_SML_DCLREC_REC] :
;  436     1686  4                BEGIN
;  437     1687  4                CH$MOVE (6, ASCII_LOGNAM, FIRQB [FQB$T_RCVNAM]);    !Logical name
;  438     1688  4                FIRQB [FQB$W_BMAX] = -1;    ! Use temporary mailboxes.
;  439     1689  4                FIRQB [FQB$B_ACCESS] = 1;   ! Local message.
;  440     1690  4                BPA$MESAG (FIRQB, 0, 0, BYTXFR);
;  441     1691  4                CH$FILL (0, 32, FIRQB [2, B_]);
;  442     1692  4                FIRQB [FQB$B_SUBFUN] = K_SML_REC;
;  443     1693  3                END;
;  444     1694  3
;  445     1695  3            [K_SML_REC] :
;  446     1696  4                BEGIN
;  447     1697  4                CH$MOVE (6, ASCII_LOGNAM, FIRQB [FQB$T_RCVNAM]);    !Logical name
;  448     1698  4                FIRQB [FQB$B_SUBFUN] = K_SML_DCLREC_REC;
;  449     1699  4                FIRQB [FQB$W_BMAX] = -1;    ! Use temporary mailboxes.
;  450     1700  4                FIRQB [FQB$B_ACCESS] = 1;   ! Local message.
;  451     1701  4                BPA$MESAG (FIRQB, 0, 0, BYTXFR);
;  452     1702  4                CH$FILL (0, 32, FIRQB [2, B_]);
;  453     1703  4                FIRQB [FQB$B_SUBFUN] = K_SML_REC;
;  454     1704  4                FIRQB [FQB$B_RMOD] = 1;     !Sleep indefinitely
;  455     1705  3                END;
;  456     1706  3            TES;
;  457     1707  3
;  458     1708  3        !+
;  459     1709  3        ! Now call BPA$MESAG, to do the work.
;  460     1710  3        !-
;  461     1711  3                BPA$MESAG (FIRQB, 0, 0, BYTXFR);
;  462     1712  3                CH$FILL (0, 30, RES_STRING);
;  463     1713  3                CH$MOVE (20, FIRQB [FQB$T_PAR_STR], RES_STRING + 8);
;  464     1714  3                STR$COPY_R (RET_STRING, %REF (30), RES_STRING);
;  465     1715  2                END;
;  466     1716  2
;  467     1717  2            [FUN$K_UUMES2] :                        ! Large message send/receive
;  468     1718  3                BEGIN
;  469     1719  3
;  470     1720  3                GLOBAL REGISTER
;  471     1721  3                    CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  472     1722  3
;  473     1723  3                LOCAL
;  474     1724  3                    BUFLEN,                     ! Length of buffer
;  475     1725  3                    BUFADR,                     ! Address of user's buffer
;  476     1726  3                    BYTXFR;                     ! Number of bytes actually transferred
;  477     1727  3
;  478     1728  3        !+
;  479     1729  3        ! Set up buffer length, byte count and buffer address based on the SYS()
```

```
;  480          1730  3 ! string.
;  481          1731  3 !-
;  482          1732  3
;  483          1733  4             IF (.FIRQB [12, B_] EQL 0)
;  484          1734  3             THEN
;  485          1735  4                 BEGIN
;  486          1736  4 !+
;  487          1737  4 ! The buffer is in the string.
;  488          1738  4 !-
;  489          1739  4                 CCB = 0;
;  490          1740  4
;  491          1741  5                 IF (.CODE_STR [DSC$W_LENGTH] GTR 40)
;  492          1742  4                 THEN
;  493          1743  5                     BEGIN
;  494          1744  5                     BUFADR = FIRQB [42, B_];
;  495          1745  5                     BUFLEN = .CODE_STR [DSC$W_LENGTH] - 40;
;  496          1746  5                     END
;  497          1747  4                 ELSE
;  498          1748  5                     BEGIN
;  499          1749  5 !+
;  500          1750  5 ! There is no buffer.
;  501          1751  5 !-
;  502          1752  5                     BUFADR = BUFLEN = 0;
;  503          1753  5                     END
;  504          1754  5
;  505          1755  4                 END
;  506          1756  3             ELSE
;  507          1757  4                 BEGIN
;  508          1758  4 !+
;  509          1759  4 ! The buffer is an I/O buffer.  The low seven bits of byte 11 are the
;  510          1760  4 ! channel number.
;  511          1761  4 !-
;  512          1762  4
;  513          1763  4                 LOCAL
;  514          1764  4                     CHAN;
;  515          1765  4
;  516          1766  4                 CHAN = (.FIRQB [12, B_] AND 127);
;  517          1767  4
;  518          1768  4                 IF (.CHAN LEQ 0) THEN BAS$$STOP_IO (BAS$K_ILLIO_CHA);
;  519          1769  4
;  520          1770  4                 BAS$$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
;  521          1771  4
;  522          1772  4                 IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  523          1773  4
;  524          1774  5                 IF ((.FIRQB [16, W_] + .FIRQB [14, W_]) GTRU .CCB [LUB$W_RBUF_SIZE])
;  525          1775  4                 THEN
;  526          1776  4                     BAS$$STOP_IO (BAS$K_ILLBYTCOU);
;  527          1777  4
;  528          1778  4                 IF (.FIRQB [14, W_] EQL 0) THEN BAS$$STOP_IO (BAS$K_NO_ROOUSE);
;  529          1779  4
;  530          1780  4                 BUFADR = .CCB [LUB$A_RBUF_ADR] + .FIRQB [16, W_];
;  531          1781  4                 BUFLEN = .FIRQB [14, W_];
;  532          1782  3                 END;
;  533          1783  3
;  534          1784  3 !+
;  535          1785  3 ! Copy the user's parameter string to the FIRQB.
;  536          1786  3 !-
```

```
;  537          1787  3                CH$MOVE (20, FIRQB [22, B_], FIRQB [12, B_]);
;  538          1788  !+
;  539          1789  ! Now do the RSTS/E .MESAG function
;  540          1790  !-
;  541          1791                   BPA$MESAG (FIRQB, .BUFLEN, .BUFADR, BYTXFR);
;  542          1792  !+
;  543          1793  ! Put things back.
;  544          1794  !-
;  545          1795                   CH$MOVE (20, FIRQB [12, B_], FIRQB [22, B_]);
;  546          1796                   FIRQB [14, W_] = .BYTXFR;
;  547          1797
;  548          1798                   IF (.CCB NEQA 0) THEN BAS$$CB_POP ();
;  549          1799
;  550          1800                   STR$COPY_R (RET_STRING, %REF (40), .FIRQB_DESC [DSC$A_POINTER]);
;  551          1801                   END;
;  552          1802
;  553          1803           [FUN$K_UUFSS1, FUN$K_UUFSS2] :          ! File string scan
;  554          1804                   BEGIN
;  555          1805
;  556          1806                   LOCAL
;  557          1807                       STATUS,
;  558          1808                       FSB : $FSB_DEF;
;  559          1809
;  560          1810                   STATUS = BPA$FSS (FIRQB, FSB, .CODE_STR [DSC$A_POINTER] + 2, .CODE_STR [DSC$W_LENGTH] - 2);
;  561          1811
;  562          1812  4                IF ( NOT .STATUS)
;  563          1813  3                THEN
;  564          1814                       LIB$STOP (.STATUS)
;  565          1815  3                ELSE
;  566          1816  4                    BEGIN
;  567          1817  4
;  568          1818  4                    LOCAL
;  569          1819  4                        STATUS;
;  570          1820  4
;  571          1821  4                    STATUS = BPA$FREE_BLOCK (.FSB [FSB$A_FSA], NAM$C_MAXRSS);
;  572          1822  4
;  573          1823  4                    IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
;  574          1824  4
;  575          1825  4  !+
;  576          1826  4  ! Return information to the user from the FIRQB and FSB.
;  577          1827  4  !-
;  578          1828  4                    FIRQB [3, B_] = .FIRQB [5, B_];
;  579          1829  4                    FIRQB [16, W_] = .FIRQB [28, W_];
;  580          1830  4                    FIRQB [20, W_] = .FIRQB [30, W_];
;  581          1831  4                    FIRQB [28, W_] = .FSB [FSB$W_FLAG_1];
;  582          1832  4                    FIRQB [30, W_] = .FSB [FSB$W_FLAG_2];
;  583          1833  4                    STR$COPY_R (RET_STRING, %REF (30), .FIRQB_DESC [DSC$A_POINTER]);
;  584          1834  3                    END;
;  585          1835  3
;  586          1836  2                END;
;  587          1837  2
;  588          1838  2           [FUN$K_UUPRI] :                         ! Set priority, etc.
;  589          1839  2  !+
;  590          1840  2  ! Only priority setting is implemented; all else is ignored.
;  591          1841  2  !-
;  592          1842  3                BEGIN
;  593          1843  3                BPA$SET_PRI (FIRQB);
```

```
;  594          1844  2              END;
;  595          1845  2
;  596          1846  2          [FUN$K_UUATR] :                         ! Read/write file
;  597          1847  2              BAS$$STOP (BAS$K_MISSPEFEA);
;  598          1848  2
;  599          1849  2          [FUN$K_UUASS] :                         ! Assign
;  600          1850  3              BEGIN
;  601          1851  3              BPA$ASSIGN (FIRQB);
;  602          1852  3              STR$COPY_DX (RET_STRING, FIRQB_DESC);
;  603          1853  2              END;
;  604          1854  2
;  605          1855  2          [FUN$K_UUDEA] :                         ! Deassign
;  606          1856  3              BEGIN
;  607          1857  3              BPA$DEASSIGN (FIRQB);
;  608          1858  2              END;
;  609          1859  2
;  610          1860  2          [FUN$K_UUDAL] :                         ! Deassign all
;  611          1861  2              BPA$DEASS_ALL ();
;  612          1862  2
;  613          1863  2          [FUN$K_UUSDEF] :                        ! Set default
;  614          1864  3              BEGIN
;  615          1865  3              BPA$SET_DEF (FIRQB);
;  616          1866  2              END;
;  617          1867  2
;  618          1868  2          [INRANGE] :                             ! Unimplemented
;  619          1869  2              BAS$$STOP (BAS$K_MISSPEFEA);
;  620          1870  2
;  621          1871  2          [OUTRANGE] :                            ! Not defined
;  622          1872  2              BAS$$STOP (BAS$K_ILLSYSUSA);
;  623          1873  2          TES;
;  624          1874  2
;  625          1875  2      STR$COPY_DX (.RESULT_STR, RET_STRING);
;  626          1876  2      STR$FREE1_DX (RET_STRING);
;  627          1877  2      RETURN;
;  628          1878  1      END;                                        ! of routine BAS$$UUO


                                                00  00  00138 P.AAA:  .BYTE   0, 0                                          ;

                                                                       .EXTRN  R50ASC

                                                    0FFC 00000 BAS$$UUO:
                                                                       .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1559
                                                5A 00000000G   00   9E 00002          MOVAB   BPA$MESAG, R10                ;
                                                59 00000000G   00   9E 00009          MOVAB   BAS$$STOP_IO, R9              ;
                                                5E      FD7C   CE   9E 00010          MOVAB   -644(SP), SP                  ;
                                          48    AE 01020229    8F   D0 00015          MOVL    #16908841, FIRQB_DESC         ; 1607
                                          4C    AE       52    AE   9E 0001D          MOVAB   FIRQB+2, FIRQB_DESC+4         ; 1610
                                                56       08    AC   D0 00022          MOVL    CODE_STR, R6                  ; 1611
    0229    8F          00        04    B6                 66   2C 00026          MOVC5   (R6), @4(R6), #0, #553, FIRQB+2       ;
                                                      52    AE        0002E
                                          F8    AD 02000000    8F   D0 00030          MOVL    #33554432, RET_STRING         ; 1612
                                                      FC    AD   D4 00038          CLRL    RET_STRING+4                  ; 1615
                                    36    E6    8F          53   AE   8F 0003B      CASEB   FIRQB+3, #-26, #54            ; 1620
          0214          02C9          02C9          02C9          00041 1$:       .WORD   32$-1$,-                      ;
          02C9          02C9          02C9          02C9          00049           .WORD   32$-1$,-                      ;
```

```
                02C9    02C9    02C9    02C9    00051                   32$-1$,-
                02C9    02C9    0283    02C9    00059                   21$-1$,-
                0075    02C9    02C9    0214    00061                   32$-1$,-
                02C9    02C9    02C9    02C9    00069                   32$-1$,-
                02C9    02C9    02C9    02C9    00071                   32$-1$,-
                02C9    02C9    02C9    02C9    00079                   32$-1$,-
                007E    02C9    02C9    02C9    00081                   32$-1$,-
                02C9    02B4    02A8    028F    00089                   32$-1$,-
                02C9    02C9    02C9    02C9    00091                   32$-1$,-
                02C9    02C9    02C9    00C2    00099                   32$-1$,-
                02C9    02C9    02C9    0173    000A1                   32$-1$,-
                        02BD    02C9    02C9    000A9                   26$-1$,-
                                                                        32$-1$,-
                                                                        32$-1$,-
                                                                        21$-1$,-
                                                                        32$-1$,-
                                                                        32$-1$,-
                                                                        2$-1$,-
                                                                        32$-1$,-
                                                                        32$-1$,-
                                                                        32
```

```
                                40   AE 020E0000    8F   D0 000BF 3$:      MOVL    #34471936, ERR_STRING                    ; 1633
                                44   AE             D4   000C7           CLRL    ERR_STRING+4                             ; 1636
                                7E   54   AE        9A   000CA           MOVZBL  FIRQB+4, -(SP)                           ; 1637
                                44   AE             9F   000CE           PUSHAB  ERR_STRING                               ;
                  00000000G     00                  02   FB 000D1           CALLS   #2, BAS$ERT                              ;
                                38   AE 01020002    8F   D0 000D8           MOVL    #16908290, HEADER                        ; 1638
                                3C   AE      FF1A   CF   9E 000E0           MOVAB   P.AAA, HEADER+4                          ; 1641
                                40   AE             9F   000E6           PUSHAB  ERR_STRING                               ; 1642
                                3C   AE             9F   000E9           PUSHAB  HEADER                                   ;
                                F8   AD             9F   000EC           PUSHAB  RET_STRING                               ;
                  00000000G     00                  03   FB 000EF           CALLS   #3, STR$CONCAT                           ;
                                40   AE             9F   000F6           PUSHAB  ERR_STRING                               ; 1643
                  00000000G     00                  01   FB 000F9           CALLS   #1, STR$FREE1_DX                         ;
                                     0212           31   00100 4$:         BRW     34$                                      ; 1620
                                40   AE             9F   00103 5$:         PUSHAB  ASCII_LOGNAM                             ; 1664
                                5A   AE             9F   00106           PUSHAB  FIRQB+6                                  ;
                                08   AE             06   D0 00109         MOVL    #6, 8(SP)                                ;
                                08   AE             9F   0010D           PUSHAB  8(SP)                                    ;
                  00000000G     00                  03   FB 00110           CALLS   #3, R50ASC                               ;
                                3F        40   AE    91   00117           CMPB    ASCII_LOGNAM, #63                        ; 1666
                                                07   12   0011B           BNEQ    6$                                       ;
                                7E        00G  8F    9A   0011D           MOVZBL  #BAS$K_ILLSYSUSA, -(SP)                  ;
                                69                   01   FB 00121           CALLS   #1, BAS$$STOP_IO                         ;
                     03         FF   8F        54   AE   8F 00124 6$:       CASEB   FIRQB+4, #-1, #3                         ; 1672
        0042              001C            006C            0008    0012A 7$:       .WORD   8$-7$,-                          ;
                                                                                    11$-7$,-                                 ;
                                                                                    9$-7$,-                                  ;
                                                                                    10$-7$                                   ;
                  0C    AE        5A   AE                14   28 00132 8$:       MOVC3   #20, FIRQB+10, MESAG             ; 1677
                  5C    AE        0C   AE                14   28 00138           MOVC3   #20, MESAG, FIRQB+12             ; 1678
                  56    AE        40   AE                06   28 0013E           MOVC3   #6, ASCII_LOGNAM, FIRQB+6        ; 1679
                                                        50   11 00144           BRB     11$                              ; 1672
                  56    AE        40   AE                06   28 00146 9$:       MOVC3   #6, ASCII_LOGNAM, FIRQB+6        ; 1687
                                5E   AE                01   AE 0014C           MNEGW   #1, FIRQB+14                     ; 1688
                                5D   AE                01   90 00150           MOVB    #1, FIRQB+13                     ; 1689
                                               04   AE   9F   00154           PUSHAB  BYTXFR                           ; 1690
                                                    7E   7C 00157           CLRQ    -(SP)                            ;
                                               5C   AE   9F   00159           PUSHAB  FIRQB                            ;
                                          6A             04   FB 0015C           CALLS   #4, BPA$MESAG                    ;
        20              00                6E             00   2C 0015F           MOVC5   #0, (SP), #0, #32, FIRQB+2       ; 1691
                                               52   AE        00164
                                54   AE                02   90 00166           MOVB    #2, FIRQB+4                      ; 1692
                                                        2A   11 0016A           BRB     11$                              ; 1672
                  56    AE        40   AE                06   28 0016C 10$:      MOVC3   #6, ASCII_LOGNAM, FIRQB+6        ; 1697
                                54   AE                01   90 00172           MOVB    #1, FIRQB+4                      ; 1698
                                5E   AE                01   AE 00176           MNEGW   #1, FIRQB+14                     ; 1699
                                5D   AE                01   90 0017A           MOVB    #1, FIRQB+13                     ; 1700
                                               04   AE   9F   0017E           PUSHAB  BYTXFR                           ; 1701
                                                    7E   7C 00181           CLRQ    -(SP)                            ;
                                               5C   AE   9F   00183           PUSHAB  FIRQB                            ;
                                          6A             04   FB 00186           CALLS   #4, BPA$MESAG                    ;
        20              00                6E             00   2C 00189           MOVC5   #0, (SP), #0, #32, FIRQB+2       ; 1702
                                               52   AE        0018E
                                54   AE      0102      8F   B0 00190           MOVW    #258, FIRQB+4                    ; 1703
                                               04   AE   9F   00196 11$:        PUSHAB  BYTXFR                           ; 1711
                                                    7E   7C 00199           CLRQ    -(SP)                            ;
                                               5C   AE   9F   0019B           PUSHAB  FIRQB                            ;
```

```
                                6A              04  FB 0019E          CALLS   #4, BPA$MESAG
        1E      00              6E              00  2C 001A1          MOVC5   #0, (SP), #0, #30, RES_STRING                ;1712
                                            20  AE    001A6
                28  AE      5C  AE              14  28 001AB          MOVC3   #20, FIRQB+12, RES_STRING+8                  ;1713
                                            20  AE  9F 001AE          PUSHAB  RES_STRING                                   ;1714
                                          00FD  31 001B1              BRW     24$
                                            5C  AE  95 001B4 12$:     TSTB    FIRQB+12                                     ;1733
                                                17  12 001B7         BNEQ    14$
                                                5B  D4 001B9         CLRL    CCB                                          ;1739
                                28              66  B1 001BB         CMPW    (R6), #40                                    ;1741
                                                0C  1B 001BE         BLEQU   13$
                                57      7A      AE  9E 001C0         MOVAB   FIRQB+42, BUFADR                             ;1744
                                58              66  3C 001C4         MOVZWL  (R6), BUFLEN                                 ;1745
                                58              28  C2 001C7         SUBL2   #40, BUFLEN
                                                58  11 001CA         BRB     19$                                          ;1741
                                                57  7C 001CC 13$:     CLRQ    BUFADR                                       ;1752
                                                54  11 001CE         BRB     19$                                          ;1741
        52      5C  AE                  07              00  EF 001D0 14$:     EXTZV   #0, #7, FIRQB+12, CHAN               ;1766
                                                07  14 001D6         BGTR    15$                                          ;1768
                                7E      00G     8F  9A 001D8         MOVZBL  #BAS$K_ILLIO_CHA, -(SP)
                                69              01  FB 001DC         CALLS   #1, BA$$$STOP_IO
                                                50  D4 001DF 15$:     CLRL    R0                                           ;1770
                                        00000000G   00  16 001E1         JSB     BAS$$CB_PUSH
                                07      FC      AB  E8 001E7         BLBS    -4(CCB), 16$                                 ;1772
                                7E      00G     8F  9A 001EB         MOVZBL  #BAS$K_IO_CHANOT, -(SP)
                                69              01  FB 001EF         CALLS   #1, BA$$$STOP_IO
                                50      60      AE  3C 001F2 16$:     MOVZWL  FIRQB+16, R0                                 ;1774
                                51      5E      AE  3C 001F6         MOVZWL  FIRQB+14, R1
                                50              51  C0 001FA         ADDL2   R1, R0
        50      D2  AB                  10              00  ED 001FD         CMPZV   #0, #16, -46(CCB), R0
                                                07  1E 00203         BGEQU   17$
                                7E      00G     8F  9A 00205         MOVZBL  #BAS$K_ILLBYTCOU, -(SP)                      ;1776
                                69              01  FB 00209         CALLS   #1, BA$$$STOP_IO
                                            5E  AE  B5 0020C 17$:     TSTW    FIRQB+14                                     ;1778
                                                07  12 0020F         BNEQ    18$
                                7E      00G     8F  9A 00211         MOVZBL  #BAS$K_NO_ROOUSE, -(SP)
                                69              01  FB 00215         CALLS   #1, BA$$$STOP_IO
                                57      60      AE  3C 00218 18$:     MOVZWL  FIRQB+16, BUFADR                             ;1780
                                57      EC      AB  C0 0021C         ADDL2   -20(CCB), BUFADR
                                58      5E      AE  3C 00220         MOVZWL  FIRQB+14, BUFLEN                             ;1781
        5C  AE          66      AE              14  28 00224 19$:     MOVC3   #20, FIRQB+22, FIRQB+12                      ;1787
                                            08  AE  9F 0022A         PUSHAB  BYTXFR                                       ;1791
                                                57  DD 0022D         PUSHL   BUFADR
                                                58  DD 0022F         PUSHL   BUFLEN
                                            5C  AE  9F 00231         PUSHAB  FIRQB
                                6A              04  FB 00234         CALLS   #4, BPA$MESAG
        66  AE          5C      AE              14  28 00237         MOVC3   #20, FIRQB+12, FIRQB+22                      ;1795
                                5E      AE      08  AE  B0 0023D         MOVW    BYTXFR, FIRQB+14                             ;1796
                                                5B  D5 00242         TSTL    CCB                                          ;1798
                                                06  13 00244         BEQL    20$
                                        00000000G   00  16 00246         JSB     BAS$$CB_POP
                                            4C  AE  DD 0024C 20$:     PUSHL   FIRQB_DESC+4                                 ;1800
                                04      AE              28  D0 0024F         MOVL    #40, 4(SP)
                                                60  11 00253         BRB     25$
                                7E                      66  3C 00255 21$:     MOVZWL  (R6), -(SP)                          ;1810
                                6E                      02  C2 00258         SUBL2   #2, (SP)
            7E          04      A6                      02  C1 0025B         ADDL3   #2, 4(R6), -(SP)
```

```
                 2C    AE  9F 00260          PUSHAB  FSB                                     ;
                 5C    AF  9F 00263          PUSHAB  FIRQB                                   ;
00000000G  00          04  FB 00266          CALLS   #4, BPA$FSS                             ;
                 0B        50  E8 0026D          BLBS    STATUS, 22$                         ; 1812
                           50  DD 00270          PUSHL   STATUS                              ; 1814
00000000G  00              01  FB 00272          CALLS   #1, LIB$STOP                        ;
                           78  11 00279          BRB     29$                                 ;
                 7E    FF  8F  9A 0027B 22$:     MOVZBL  #255, -(SP)                         ; 1821
                 2C    AE  DD 0027F          PUSHL   FSB+4                               ;
00000000G  00              02  FB 00282          CALLS   #2, BPA$FREE_BLOCK                  ;
                 09        50  E8 00289          BLBS    STATUS, 23$                         ; 1823
                           50  DD 0028C          PUSHL   STATUS                              ;
00000000G  00              01  FB 0028E          CALLS   #1, LIB$STOP                        ;
       53  AE    55  AE  90 00295 23$:     MOVB    FIRQB+5, FIRQB+3                    ; 1828
       60  AE    6C  AE  B0 0029A          MOVW    FIRQB+28, FIRQB+16                  ; 1829
       64  AE    6E  AE  B0 0029F          MOVW    FIRQB+30, FIRQB+20                  ; 1830
       6C  AE    46  AE  B0 002A4          MOVW    FSB+34, FIRQB+28                    ; 1831
       6E  AE    44  AE  B0 002A9          MOVW    FSB+32, FIRQB+30                    ; 1832
                 4C    AE  DD 002AE          PUSHL   FIRQB_DESC+4                        ; 1833
       04  AE              1E  D0 002B1 24$:     MOVL    #30, 4(SP)                          ;
                 04    AE  9F 002B5 25$:     PUSHAB  4(SP)                               ;
                 F8    AD  9F 002B8          PUSHAB  RET_STRING                          ;
00000000G  00              03  FB 002BB          CALLS   #3, STR$COPY_R                      ;
                           51  11 002C2          BRB     34$                                 ; 1620
                 50    AE  9F 002C4 26$:     PUSHAB  FIRQB                               ; 1843
00000000G  00              01  FB 002C7          CALLS   #1, BPA$SET_PRI                     ;
                           45  11 002CE          BRB     34$                                 ; 1620
                 50    AE  9F 002D0 27$:     PUSHAB  FIRQB                               ; 1851
00000000G  00              01  FB 002D3          CALLS   #1, BPA$ASSIGN                      ;
                 48    AE  9F 002DA          PUSHAB  FIRQB_DESC                          ; 1852
                 F8    AD  9F 002DD          PUSHAB  RET_STRING                          ;
00000000G  00              02  FB 002E0          CALLS   #2, STR$COPY_DX                     ;
                           2C  11 002E7          BRB     34$                                 ; 1620
                 50    AE  9F 002E9 28$:     PUSHAB  FIRQB                               ; 1857
00000000G  00              01  FB 002EC          CALLS   #1, BPA$DEASSIGN                    ;
                           20  11 002F3 29$:     BRB     34$                                 ; 1620
00000000G  00              00  FB 002F5 30$:     CALLS   #0, BPA$DEASS_ALL                   ; 1861
                           17  11 002FC          BRB     34$                                 ;
                 50    AE  9F 002FE 31$:     PUSHAB  FIRQB                               ; 1865
00000000G  00              01  FB 00301          CALLS   #1, BPA$SET_DEF                     ;
                           0B  11 00308          BRB     34$                                 ; 1620
                 7E        00G 8F  9A 0030A 32$:     MOVZBL  #BAS$K_MISSPEFEA, -(SP)             ; 1869
00000000G  00              01  FB 0030E 33$:     CALLS   #1, BAS$$STOP                       ;
                 F8    AD  9F 00315 34$:     PUSHAB  RET_STRING                          ; 1875
                 04    AC  DD 00318          PUSHL   RESULT_STR                          ;
00000000G  00              02  FB 0031B          CALLS   #2, STR$COPY_DX                     ;
                 F8    AD  9F 00322          PUSHAB  RET_STRING                          ; 1876
00000000G  00              01  FB 00325          CALLS   #1, STR$FREE1_DX                    ;
                               04  0032C          RET                                     ; 1878

; Routine Size:  813 bytes,    Routine Base:  _BAS$CODE + 013A

;  629          1879  1 END                                    ! of module BAS$SYS
;  630          1880  1
;  631          1881  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _BAS$CODE                      1127  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                                Library Statistics
;
;                                          --------  Symbols  --------      Pages       Processing
;       File                               Total     Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1        9776        13         0         581        00:01.2



;                                  COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASSYS/OBJ=OBJ$:BASSYS MSRC$:BASSYS/UPDATE=(ENH$:BASSYS)

; Size:         1125 code + 2 data bytes
; Run Time:       00:25.7
; Elapsed Time:   00:58.6
; Lines/CPU Min:   4388
; Lexemes/CPU-Min: 31558
; Memory Used:  288 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASUDFWF
LIS

BASSTRING
LIS

BASTERMIO
LIS

BASTRM
LIS

BASTAB
LIS

BASUDFRM
LIS

BASSYS
LIS

BASSTR
LIS

BASUDFRL
LIS