```
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   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  SSSSSSSSSSS      RRR         RRR   TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS      RRR         RRR   TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS      RRR         RRR   TTT        LLLLLLLLLLLLLLL
```

```
BBBBBBBB      AAAAAA      SSSSSSSS  BBBBBBBB    UU      UU  FFFFFFFFFF    SSSSSSSS    IIIIII    ZZZZZZZZZZ
BBBBBBBB      AAAAAA      SSSSSSSS  BBBBBBBB    UU      UU  FFFFFFFFFF    SSSSSSSS    IIIIII    ZZZZZZZZZZ
BB      BB   AA      AA  SS        BB      BB  UU      UU  FF            SS            II            ZZ
BB      BB   AA      AA  SS        BB      BB  UU      UU  FF            SS            II            ZZ
BB      BB   AA      AA  SS        BB      BB  UU      UU  FF            SS            II          ZZ
BB      BB   AA      AA  SS        BB      BB  UU      UU  FF            SS            II          ZZ
BBBBBBBB     AA      AA    SSSSSS  BBBBBBBB    UU      UU  FFFFFFFF      SSSSSS      II        ZZ
BBBBBBBB     AA      AA    SSSSSS  BBBBBBBB    UU      UU  FFFFFFFF      SSSSSS      II        ZZ
BB      BB   AAAAAAAAAA          SS  BB      BB  UU      UU  FF                  SS    II      ZZ
BB      BB   AAAAAAAAAA          SS  BB      BB  UU      UU  FF                  SS    II      ZZ
BB      BB   AA      AA          SS  BB      BB  UU      UU  FF                  SS    II    ZZ                ....
BB      BB   AA      AA          SS  BB      BB  UU      UU  FF                  SS    II    ZZ                ....
BBBBBBBB     AA      AA  SSSSSSSS    BBBBBBBB    UUUUUUUUUU  FF            SSSSSSSS    IIIIII  ZZZZZZZZZZ      ....
BBBBBBBB     AA      AA  SSSSSSSS    BBBBBBBB    UUUUUUUUUU  FF            SSSSSSSS    IIIIII  ZZZZZZZZZZ      ....
```

```
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$BUFSIZ (                          ! Get buffer size
;     2          0002  0                     IDENT = '1-003'          ! File: BASBUFSIZ.B32, Edit: JBS1003
;     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 Miscellaneous I/O
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains the BASIC BUFSIZ function, which returns
;    36          0036  1 !       the buffer size of the file open on the specified channel.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  VAX-11 User Mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: 11-APR-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.
;    45          0045  1 ! 1-002 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
;    46          0046  1 ! 1-003 - Use channel 0, not device TT.
;    47          0047  1 !         JBS 11-MAR-1980
;    48          0048  1 !--
;    49          0049  1
;    50          0050  1 !<BLF/PAGE>
```

```
;    52          0051  1 !
;    53          0052  1 ! SWITCHES:
;    54          0053  1 !
;    55          0054  1
;    56          0055  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    57          0056  1
;    58          0057  1 !
;    59          0058  1 ! LINKAGES:
;    60          0059  1 !
;    61          0060  1
;    62          0061  1 REQUIRE 'RTLIN:OTSLNK';                         ! Define linkages
;    63          0490  1
;    64          0491  1 !
;    65          0492  1 ! TABLE OF CONTENTS:
;    66          0493  1 !
;    67          0494  1
;    68          0495  1 FORWARD ROUTINE
;    69          0496  1     BAS$BUFSIZ;                                 ! Return buffer size
;    70          0497  1
;    71          0498  1 !
;    72          0499  1 ! INCLUDE FILES:
;    73          0500  1 !
;    74          0501  1
;    75          0502  1 REQUIRE 'RTLML:OTSLUB';                         ! Get LUB definitions
;    76          0642  1
;    77          0643  1 REQUIRE 'RTLML:OTSISB';                         ! Get ISB definitions
;    78          0811  1
;    79          0812  1 REQUIRE 'RTLIN:RTLPSECT';                       ! Macros for defining psects
;    80          0907  1
;    81          0908  1 LIBRARY 'RTLSTARLE';                            ! System symbols
;    82          0909  1
;    83          0910  1 !
;    84          0911  1 ! MACROS:
;    85          0912  1 !
;    86          0913  1 !       NONE
;    87          0914  1 !
;    88          0915  1 ! EQUATED SYMBOLS:
;    89          0916  1 !
;    90          0917  1 !       NONE
;    91          0918  1 !
;    92          0919  1 ! PSECTS:
;    93          0920  1 !
;    94          0921  1 DECLARE_PSECTS (BAS);                           ! Declare psects for BAS$ facility
;    95          0922  1 !
;    96          0923  1 ! OWN STORAGE:
;    97          0924  1 !
;    98          0925  1 !       NONE
;    99          0926  1 !
;   100          0927  1 ! EXTERNAL REFERENCES:
;   101          0928  1 !
;   102          0929  1
;   103          0930  1 EXTERNAL ROUTINE
;   104          0931  1     BAS$$OPEN_ZERO : NOVALUE,                   ! Open channel 0
;   105          0932  1     BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,         ! Load register CCB
;   106          0933  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,           ! Done with register CCB
;   107          0934  1     BAS$$STOP : NOVALUE;                        ! Signal fatal error
;   108          0935  1
```

```
;  109          0936  1 !+
;  110          0937  1 ! The following are the error codes used in this module.
;  111          0938  1 !-
;  112          0939  1
;  113          0940  1 EXTERNAL LITERAL
;  114          0941  1     BAS$K_PROLOSSOR : UNSIGNED (8);             ! Program lost, sorry
;  115          0942  1
```

```
;  117          0943  1 GLOBAL ROUTINE BAS$BUFSIZ (                   ! Return buffer size
;  118          0944  1                 CHAN                         ! Channel whose buffer size to return
;  119          0945  1     ) =
;  120          0946  1
;  121          0947  1 !++
;  122          0948  1 ! FUNCTIONAL DESCRIPTION:
;  123          0949  1 !
;  124          0950  1 !       Returns the size of the buffer for the specified channel.
;  125          0951  1 !       If the channel is closed a zero is returned.
;  126          0952  1 !
;  127          0953  1 ! FORMAL PARAMETERS:
;  128          0954  1 !
;  129          0955  1 !       CHAN.rl.v       The channel whose buffer size to return.
;  130          0956  1 !
;  131          0957  1 ! IMPLICIT INPUTS:
;  132          0958  1 !
;  133          0959  1 !       The LUB$W_RBUF_SIZE field of the LUB of the specified channel.
;  134          0960  1 !
;  135          0961  1 ! IMPLICIT OUTPUTS:
;  136          0962  1 !
;  137          0963  1 !       NONE
;  138          0964  1 !
;  139          0965  1 ! ROUTINE VALUE:
;  140          0966  1 !
;  141          0967  1 !       The number of bytes in the buffer, as a longword integer.
;  142          0968  1 !
;  143          0969  1 ! SIDE EFFECTS:
;  144          0970  1 !
;  145          0971  1 !       Signals if an error is encountered.
;  146          0972  1 !       BAS$$CB_PUSH will signal if the channel number is invalid.
;  147          0973  1 !
;  148          0974  1 !--
;  149          0975  1
;  150          0976  2     BEGIN
;  151          0977  2
;  152          0978  2     BUILTIN
;  153          0979  2         FP;
;  154          0980  2
;  155          0981  2     GLOBAL REGISTER
;  156          0982  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  157          0983  2
;  158          0984  2     LOCAL
;  159          0985  2         BUFFER_SIZE,
;  160          0986  2         FMP : REF BLOCK [, BYTE];
;  161          0987  2
;  162          0988  2     FMP = .FP;
;  163          0989  2 !+
;  164          0990  2 ! If this is channel zero, get the user's terminal, and be sure it is open.
;  165          0991  2 !-
;  166          0992  2
;  167          0993  3     IF (.CHAN EQL 0)
;  168          0994  2     THEN
;  169          0995  3         BEGIN
;  170          0996  3         BAS$$CB_PUSH (LUB$K_LUN_INPU, LUB$K_ILUN_MIN);
;  171          0997  3         CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  172          0998  3
;  173          0999  3         IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$OPEN_ZERO (.FMP [SF$L_SAVE_FP]);
```

```
;  174          1000  3
;  175          1001  3          END
;  176          1002  2      ELSE
;  177          1003  3          BEGIN
;  178          1004  3          BAS$$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
;  179          1005  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  180          1006  2          END;
;  181          1007  2
;  182          1008  2      !+
;  183          1009  2      ! Get the buffer size from the Logical Unit Block.  This will be zero
;  184          1010  2      ! if the channel has not been opened.
;  185          1011  2      !-
;  186          1012  2          BUFFER_SIZE = .CCB [LUB$W_RBUF_SIZE];
;  187          1013  2      !+
;  188          1014  2      ! We are done with register CCB.
;  189          1015  2      !-
;  190          1016  2          BAS$$CB_POP ();
;  191          1017  2      !+
;  192          1018  2      ! All done.
;  193          1019  2      !-
;  194          1020  2          RETURN (.BUFFER_SIZE);
;  195          1021  1          END;                                    ! end of BAS$BUFSIZ

                                                                         .TITLE  BAS$BUFSIZ
                                                                         .IDENT  \1-003\

                                                                         .EXTRN  BAS$$OPEN_ZERO, BAS$$CB_PUSH
                                                                         .EXTRN  BAS$$CB_POP, BAS$$STOP
                                                                         .EXTRN  BAS$K_PROLOSSOR

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

                                    081C 00000                           .ENTRY  BAS$BUFSIZ, Save R2,R3,R4,R11                ; 0943
                          54 00000000G  00   9E 00002                    MOVAB   BAS$$CB_PUSH, R4                             ;
                          53            5D   D0 00009                    MOVL    FP, FMP                                      ; 0988
                                  04    AC   D5 0000C                    TSTL    CHAN                                         ; 0993
                                        1E   12 0000F                    BNEQ    1$                                           ;
                          50            08   CE 00011                    MNEGL   #8, R0                                       ; 0996
                          52            07   CE 00014                    MNEGL   #7, R2                                       ;
                                        64   16 00017                    JSB     BAS$$CB_PUSH                                 ;
                    FF4C  CB      0C    A3   D0 00019                    MOVL    12(FMP), -180(CCB)                           ; 0997
                          1A      FC    AB   E8 0001F                    BLBS    -4(CCB), 2$                                  ; 0999
                                  0C    A3   DD 00023                    PUSHL   12(FMP)                                      ;
                00000000G 00            01   FB 00026                    CALLS   #1, BAS$$OPEN_ZERO                           ;
                                        0E   11 0002D                    BRB     2$                                           ; 0993
                                        50   D4 0002F  1$:               CLRL    R0                                           ; 1004
                          52    04      AC   D0 00031                    MOVL    CHAN, R2                                     ;
                                        64   16 00035                    JSB     BAS$$CB_PUSH                                 ;
                    FF4C  CB      0C    A3   D0 00037                    MOVL    12(FMP), -180(CCB)                           ; 1005
                          52      D2    AB   3C 0003D  2$:               MOVZWL  -46(CCB), BUFFER_SIZE                        ; 1012
                                00000000G 00   16 00041                    JSB     BAS$$CB_POP                                  ; 1016
                          50            52   D0 00047                    MOVL    BUFFER_SIZE, R0                              ; 1020
                                        04   04 0004A                    RET                                                  ; 1021

; Routine Size:  75 bytes,    Routine Base:  _BAS$CODE + 0000
```

```
;  196          1022  1
;  197          1023  1 END                          ! end of module BAS$BUFSIZ
;  198          1024  1
;  199          1025  0 ELUDOM
```

```
;                              PSECT SUMMARY
;
;      Name                    Bytes                  Attributes
;
;  _BAS$CODE                     75  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          1         0          581        00:01.2
```

```
;                                    COMMAND QUALIFIERS

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

; Size:          75 code + 0 data bytes
; Run Time:          00:08.3
; Elapsed Time:      00:20.5
; Lines/CPU Min:      7436
; Lexemes/CPU-Min: 45155
; Memory Used:  115 pages
; Compilation Complete
```

BASERRMSG REQ

BASFRAME REQ

BASOPN REQ

BPAFSBDEF REQ

BPASTRUCT REQ

BPADABDEF REQ

BPAXRBDEF REQ

BPAERRDEF REQ

BPAFUNDEF REQ

BASINARG REQ

BASBUFSIZ LIS

MATRIX MAR

BASIOERR REQ

BASCB LIS

BPAFQBDEF REQ

BASRTL2 MAP

BASPAR SDL

BASLNK REQ

BPAMSGDEF REQ

BASRTL

BASRTL MAP

BASCANTYP LIS