```
                                                                          M  1





BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  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
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**BASFETCHA                                    G  2
```

```
BBBBBBBB      AAAAAA      SSSSSSSS  FFFFFFFFFF  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC  HH      HH    AAAAAA
BBBBBBBB      AAAAAA      SSSSSSSS  FFFFFFFFFF  EEEEEEEEEE  TTTTTTTTTT    CCCCCCCC  HH      HH    AAAAAA
BB      BB  AA      AA  SS          FF          EE              TT      CC          HH      HH  AA      AA
BB      BB  AA      AA  SS          FF          EE              TT      CC          HH      HH  AA      AA
BB      BB  AA      AA  SS          FF          EE              TT      CC          HH      HH  AA      AA
BB      BB  AA      AA  SS          FF          EE              TT      CC          HH      HH  AA      AA
BBBBBBBB    AA      AA    SSSSSS    FFFFFFFF    EEEEEEEE        TT      CC          HHHHHHHHHH  AA      AA
BBBBBBBB    AA      AA    SSSSSS    FFFFFFFF    EEEEEEEE        TT      CC          HHHHHHHHHH  AA      AA
BB      BB  AAAAAAAAAA          SS  FF          EE              TT      CC          HH      HH  AAAAAAAAAA
BB      BB  AAAAAAAAAA          SS  FF          EE              TT      CC          HH      HH  AAAAAAAAAA
BB      BB  AA      AA          SS  FF          EE              TT      CC          HH      HH  AA      AA      ....
BB      BB  AA      AA          SS  FF          EE              TT      CC          HH      HH  AA      AA      ....
BBBBBBBB    AA      AA  SSSSSSSS    FF          EEEEEEEEEE      TT        CCCCCCCC  HH      HH  AA      AA      ....
BBBBBBBB    AA      AA  SSSSSSSS    FF          EEEEEEEEEE      TT        CCCCCCCC  HH      HH  AA      AA      ....

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
```

```
0 MODULE BAS$FETCH_ADDR (                       ! Fetch address of array element
0                  IDENT = '1-004'              ! File: BASFETCHA.B32 Edit: PLL1004
0                  ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:  BASIC Language Support
1 !
1 ! ABSTRACT:
1 !
1 !       This module calculates the address of a non-virtual array
1 !       element.  It is called by the compiled code for the LOC
1 !       function and for arrays passed as parameters.
1 !
1 ! ENVIRONMENT:  VAX-11 User Mode
1 !
1 ! AUTHOR: Pamela L. Levesque, CREATION DATE: 19-FEB-1982
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  PLL 19-Feb-1982
1 ! 1-002 - Add support for decimal arrays.  This involves calculating the
1 !         size of elements in bytes (the length in the descriptor is the
1 !         number of digits not including the sign), and using that length
1 !         to calculate the linear index.  PLL 12-Mar-1982
1 ! 1-003 - Offset for 1st index is 1, not 2.  PLL 19-Mar-1982
1 ! 1-004 - Return address of descriptor for dynamic strings.  PLL 29-Mar-1982
1 !--
1
1 !<BLF/PAGE>
```

```
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1 !    NONE
1 !
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     BAS$FETCH_ADDR;                               ! Fetch address of array element
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 REQUIRE 'RTLIN:RTLPSECT';                         ! Macros for defining psects
1
1 LIBRARY 'RTLSTARLE';                              ! System symbols
1
1 !
1 ! MACROS:
1 !
1 !        NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !        NONE
1 !
1 ! PSECTS:
1 !
1 DECLARE_PSECTS (BAS);                             ! Declare psects for BAS$ facility
1 !
1 ! OWN STORAGE:
1 !
1 !        NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1 EXTERNAL ROUTINE
1     BAS$$STOP : NOVALUE;                          ! Signal fatal error
1
1 EXTERNAL LITERAL
1     BAS$K_ARGDONMAT : UNSIGNED (8),
1     BAS$K_NOTIMP : UNSIGNED (8),
1     BAS$K_SUBOUTRAN : UNSIGNED (8),
1     BAS$K_TOOFEWARG : UNSIGNED (8),
1     BAS$K_TOOMANARG : UNSIGNED (8);
1
1
```

```
GLOBAL ROUTINE BAS$FETCH_ADDR (                         ! Fetch address of array element
        DESCRIP,                                        ! The descriptor
        INDEX1                                          ! First index
    ) : =

!++
! FUNCTIONAL DESCRIPTION:
!
!       Given a descriptor for the array and the indices, calculate
!       the address of an element.  Take into account that this may
!       be a FORTRAN array.  This routine does not handle virtual
!       arrays.
!
! FORMAL PARAMETERS:
!
!       DESCRIP.rx.da   The descriptor of the array
!       INDEX1.rl.v     The first index into the array.  More indicies
!                       may follow this one in the calling sequence.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
!
!       The address of the element is returned
!
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       Signals if an error is encountered.
!
!--

    BEGIN

    BUILTIN
        ACTUALCOUNT,
        ACTUALPARAMETER;

    LOCAL
        INDEX_VALUE,
        VALUE_LOCATION,
        MULTIPLIERS : REF VECTOR,
        BOUNDS : REF VECTOR,
        LOW_INDEX,
        HIGH_INDEX,
        INDEX_INCR,
        INDEX_NUMBER,
        VALUE_ADDR,
2               LENGTH;
2
2           MAP
2               DESCRIP : REF BLOCK [8, BYTE];
2
2   !+
2   ! Be sure the number of array subscripts matches the number of
2   ! indicies given to us.
2   !-
2
3           IF ((ACTUALCOUNT () - 1) NEQU .DESCRIP [DSC$B_DIMCT])
2           THEN
3               BEGIN
3
4               IF ((ACTUALCOUNT () - 1) LSSU .DESCRIP [DSC$B_DIMCT])
3               THEN
3                   BAS$$STOP (BAS$K_TOOFEWARG)
3               ELSE
3                   BAS$$STOP (BAS$K_TOOMANARG);
3
2               END;
2
2   !+
2   ! The coefficients and bounds must be present.
2   !-
2
2           IF ( NOT (.DESCRIP [DSC$V_FL_COEFF] AND .DESCRIP [DSC$V_FL_BOUNDS])) THEN BAS$$STOP (BAS$K_ARGDONMAT);
2
2           MULTIPLIERS = DESCRIP [DSC$L_M1];
2           BOUNDS = DESCRIP [DSC$L_M1] + (%UPVAL*.DESCRIP [DSC$B_DIMCT]);
2   !+
2   ! Compute the lower and upper index numbers based on how the array
2   ! is stored.
2   !-
2
3           IF (.DESCRIP [DSC$V_FL_COLUMN])
2           THEN
3               BEGIN
3               LOW_INDEX = .DESCRIP [DSC$B_DIMCT];
3               HIGH_INDEX = 1;
3               INDEX_INCR = -1;
3               END
2           ELSE
3               BEGIN
3               LOW_INDEX = 1;
3               HIGH_INDEX = .DESCRIP [DSC$B_DIMCT];
3               INDEX_INCR = 1;
2               END;
2
2           INDEX_NUMBER = .LOW_INDEX - .INDEX_INCR;
2   !+
2   ! Recompute decimal length if necessary.
2   !-
2           IF .DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
2           THEN
2               LENGTH = .DESCRIP [DSC$W_LENGTH]/2 + 1
2           ELSE
```

```
2           LENGTH = .DESCRIP [DSC$W_LENGTH];
2   !+
2   ! Compute the linear index from the indices provided.
2   !-
2       VALUE_LOCATION = 0;
2
2       WHILE ((INDEX_NUMBER = .INDEX_NUMBER + .INDEX_INCR) NEQ (.HIGH_INDEX + .INDEX_INCR)) DO
3           BEGIN
3           INDEX_VALUE = ACTUALPARAMETER (.INDEX_NUMBER + 1);
3
5           IF ((.INDEX_VALUE LSS .BOUNDS [(.INDEX_NUMBER - 1)*2])  !
4               OR (.INDEX_VALUE GTR .BOUNDS [((.INDEX_NUMBER - 1)*2) + 1]))
3           THEN
3               BAS$$STOP (BAS$K_SUBOUTRAN);
3
3           VALUE_LOCATION = (.VALUE_LOCATION*.MULTIPLIERS [.INDEX_NUMBER - 1]) + .INDEX_VALUE;
2           END;
2
2       VALUE_LOCATION = (.VALUE_LOCATION*.LENGTH) + .DESCRIP [DSC$A_A0];
2
2   !+
2   ! Check for an array of descriptors.  Fetch the address from the pointer
2   ! field of the descriptor if necessary.
2   !-
2
3       IF (.DESCRIP [DSC$B_DTYPE] EQLU DSC$K_DTYPE_DSC)
2       THEN
3           BEGIN
3
3           MAP
3               VALUE_LOCATION : REF BLOCK [8, BYTE];
3
3           IF .VALUE_LOCATION [DSC$B_DTYPE] NEQ DSC$K_DTYPE_T
3           THEN
3               VALUE_ADDR = .VALUE_LOCATION [DSC$A_POINTER]
3           ELSE
3               VALUE_ADDR = .VALUE_LOCATION;
3           END
2       ELSE
3           BEGIN
3           VALUE_ADDR = .VALUE_LOCATION;
2           END;
2
2
2       IF (.DESCRIP [DSC$B_CLASS] NEQU DSC$K_CLASS_A) THEN BAS$$STOP (BAS$K_NOTIMP);
2
2       RETURN .VALUE_ADDR;
2
1       END;                                            ! end of BAS$FETCH_ADDR

                                                         .TITLE  BAS$FETCH_ADDR
                                                         .IDENT  \1-004\

                                                         .EXTRN  BAS$$STOP, BAS$K_ARGDONMAT
                                                         .EXTRN  BAS$K_NOTIMP, BAS$K_SUBOUTRAN
                                                         .EXTRN  BAS$K_TOOFEWARG
                                                         .EXTRN  BAS$K_TOOMANARG

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

                                     0FFC 00000          .ENTRY  BAS$FETCH_ADDR, Save R2,R3,R4,R5,R6,R7,R8,- ; 0205
                                                                 R9,R10,R11                                   ;
                      5B 00000000G  00  9E 00002          MOVAB   BAS$$STOP, R11                               ;
                      50            6C  9A 00009          MOVZBL  (AP), R0                                     ; 0272
                                    50  D7 0000C          DECL    R0                                           ;
                      56       04   AC  D0 0000E          MOVL    DESCRIP, R6                                  ;
                      52       0B   A6  9A 00012          MOVZBL  11(R6), R2                                   ;
                      52            50  D1 00016          CMPL    R0, R2                                       ;
                                    17  13 00019          BEQL    3$                                           ;
                      50            6C  9A 0001B          MOVZBL  (AP), R0                                     ; 0276
                                    50  D7 0001E             CL   R0                                           ;
                      52            50  D1 00020          CMPL    R0, R2                                       ;
                                    06  1E 00023          BGEQU   1$                                           ;
                      7E      00G   8F  9A 00025          MOVZBL  #BAS$K_TOOFEWARG, -(SP)                      ; 0278
                                    04  11 00029          BRB     2$                                           ;
                      7E      00G   8F  9A 0002B 1$:      MOVZBL  #BAS$K_TOOMANARG, -(SP)                      ; 0280
                      6B            01  FB 0002F 2$:      CALLS   #1, BAS$$STOP                                ;
      05        0A    A6            06  E1 00032 3$:      BBC     #6, 10(R6), 4$                               ; 0288
                               0A   A6  95 00037          TSTB    10(R6)                                       ;
                                    07  19 0003A          BLSS    5$                                           ;
                      7E      00G   8F  9A 0003C 4$:      MOVZBL  #BAS$K_ARGDONMAT, -(SP)                      ;
                      6B            01  FB 00040          CALLS   #1, BAS$$STOP                                ;
                      55       14   A6  9E 00043 5$:      MOVAB   20(R6), MULTIPLIERS                          ; 0290
                      57       14 A642  DE 00047          MOVAL   20(R6)[R2], BOUNDS                           ; 0291
      0B        0A    A6            05  E1 0004C          BBC     #5, 10(R6), 6$                               ; 0297
                      51            52  D0 00051          MOVL    R2, LOW_INDEX                                ; 0300
                      50            01  D0 00054          MOVL    #1, HIGH_INDEX                               ; 0301
                      58            01  CE 00057          MNEGL   #1, INDEX_INCR                               ; 0302
                                    09  11 0005A          BRB     7$                                           ; 0297
                      51            01  D0 0005C 6$:      MOVL    #1, LOW_INDEX                                ; 0306
                      50            52  D0 0005F          MOVL    R2, HIGH_INDEX                               ; 0307
                      58            01  D0 00062          MOVL    #1, INDEX_INCR                               ; 0308
      52              51            58  C3 00065 7$:      SUBL3   INDEX_INCR, LOW_INDEX, INDEX_NUMBER          ; 0311
                      15       02   A6  91 00069          CMPB    2(R6), #21                                   ; 0315
                                    0A  12 0006D          BNEQ    8$                                           ;
                      54            66  3C 0006F          MOVZWL  (R6), R4                                     ; 0317
                      54            02  C6 00072          DIVL2   #2, R4                                       ;
                                    54  D6 00075          INCL    LENGTH                                       ;
                                    03  11 00077          BRB     9$                                           ;
                      54            66  3C 00079 8$:      MOVZWL  (R6), LENGTH                                 ; 0319
                                    53  D4 0007C 9$:      CLRL    VALUE_LOCATION                               ; 0323
      5A              50            58  C1 0007E          ADDL3   INDEX_INCR, HIGH_INDEX, R10                  ; 0325
                      52            58  C0 00082 10$:     ADDL2   INDEX_INCR, INDEX_NUMBER                     ;
                      5A            52  D1 00085          CMPL    INDEX_NUMBER, R10                            ;
                                    2A  13 00088          BEQL    13$                                          ;
                      59       04 AC42  D0 0008A          MOVL    4(AP)[INDEX_NUMBER], INDEX_VALUE             ; 0327
      50              52            01  78 0008F          ASHL    #1, INDEX_NUMBER, R0                         ; 0329
                F8 A740             59  D1 00093          CMPL    INDEX_VALUE, -8(BOUNDS)[R0]                  ;
                                    07  19 00098          BLSS    11$                                          ;
                FC A740             59  D1 0009A          CMPL    INDEX_VALUE, -4(BOUNDS)[R0]                  ; 0330
                                    07  15 0009F          BLEQ    12$                                          ;
                      7E      00G   8F  9A 000A1 11$:     MOVZBL  #BAS$K_SUBOUTRAN, -(SP)                      ; 0332
                      6B            01  FB 000A5          CALLS   #1, BAS$$STOP                                ;
```

```
                 50           53        FC A542  C5 000A8 12$:    MULL3    -4(MULTIPLIERS)[INDEX_NUMBER], -                      ; 0334
                                                                           VALUE_LOCATION, R0                                    ;
                 53           50             59  C1 000AE         ADDL3    INDEX_VALUE, R0, VALUE_LOCATION                       ;
                                             CE  11 000B2         BRB      10$                                                   ; 0325
                 50           53             54  C5 000B4 13$:    MULL3    LENGTH, VALUE_LOCATION, R0                            ; 0337
                 53           50        10   A6  C1 000B8         ADDL3    16(R6), R0, VALUE_LOCATION                            ;
                              18        02   A6  91 000BD         CMPB     2(R6), #24                                            ; 0344
                                             0C  12 000C1         BNEQ     14$                                                   ;
                              0E        02   A3  91 000C3         CMPB     2(VALUE_LOCATION), #14                                ; 0351
                                             06  13 000C7         BEQL     14$                                                   ;
                              52        04   A3  D0 000C9         MOVL     4(VALUE_LOCATION), VALUE_ADDR                         ; 0353
                                             03  11 000CD         BRB      15$                                                   ;
                              52             53  D0 000CF 14$:    MOVL     VALUE_LOCATION, VALUE_ADDR                            ; 0359
                              04        03   A6  91 000D2 15$:    CMPB     3(R6), #4                                             ; 0363
                                             07  13 000D6         BEQL     16$                                                   ;
                              7E        00G  8F  9A 000D8         MOVZBL   #BAS$K_NOTIMP, -(SP)                                  ;
                              6B             01  FB 000DC         CALLS    #1, BAS$$STOP                                         ;
                              50             52  D0 000DF 16$:    MOVL     VALUE_ADDR, R0                                        ; 0365
                                                 04 000E2         RET                                                            ; 0367

; Routine Size:  227 bytes,    Routine Base:  _BAS$CODE + 0000


;  276            0368  1
;  277            0369  1 END                                                   ! end of module BAS$FETCH_ADDR
;  278            0370  1
;  279            0371  0 ELUDOM




;                                      PSECT SUMMARY
;
;        Name                       Bytes                        Attributes
;
;  _BAS$CODE                            227  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       14         0          581          00:01.1




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

; Size:          227 code + 0 data bytes
; Run Time:         00:07.2
; Elapsed Time:     00:16.2
; Lines/CPU Min:     3078
; Lexemes/CPU-Min: 15585
; Memory Used:  103 pages
; Compilation Complete
```

0023 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASFREE
LIS

BASEXITHA
LIS

BASFETCHD
LIS

BASFORINT
LIS

BASGETRFA
LIS

BASFETCHA
LIS

BASGET
LIS

BASFSP
LIS

BASFIND
LIS

BASFORMAT
LIS

BASHANDLE
LIS