```
M 1
```

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     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     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           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**BASMATIO                                   G  4
```

```
BBBBBBBB      AAAAAA      SSSSSSSS  MM        MM    AAAAAA    TTTTTTTTTT    IIIIII      000000
BBBBBBBB      AAAAAA      SSSSSSSS  MM        MM    AAAAAA    TTTTTTTTTT    IIIIII      000000
BB      BB  AA      AA  SS          MMMM    MMMM  AA      AA      TT          II      00      00
BB      BB  AA      AA  SS          MMMM    MMMM  AA      AA      TT          II      00      00
BB      BB  AA      AA  SS          MM  MM  MM    AA      AA      TT          II      00      00
BB      BB  AA      AA  SS          MM  MM  MM    AA      AA      TT          II      00      00
BBBBBBBB    AA      AA    SSSSSS    MM        MM  AA      AA      TT          II      00      00
BBBBBBBB    AA      AA    SSSSSS    MM        MM  AA      AA      TT          II      00      00
BB      BB  AAAAAAAAAA          SS  MM        MM  AAAAAAAAAA      TT          II      00      00
BB      BB  AAAAAAAAAA          SS  MM        MM  AAAAAAAAAA      TT          II      00      00
BB      BB  AA      AA          SS  MM        MM  AA      AA      TT          II      00      00      ....
BB      BB  AA      AA          SS  MM        MM  AA      AA      TT          II      00      00      ....
BBBBBBBB    AA      AA  SSSSSSSS    MM        MM  AA      AA      TT        IIIIII      000000        ....
BBBBBBBB    AA      AA  SSSSSSSS    MM        MM  AA      AA      TT        IIIIII      000000        ....

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$MAT_IO (                                     !Basic Matrix I/O element transmitter - UPI level
0                 IDENT = '1-016'                         ! File: BASMATIO.B32 Edit: DG1016
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: VAX-11 BASIC Language support
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the UPI level element transmitters for Matrix I/O.
1 !       For matrix I/O, an element is an entire array.  The UPI then marches
1 !       through the descriptor and calls the PRINT or INPUT UDF for each element
1 !       in the array.  MAT I/O gets its own UPI module so that MAT I/O can be
1 !       excluded from the sharable library.
1 !
1 ! ENVIRONMENT: User access mode, AST reentrant.
1 !
1 ! AUTHOR: Donald G. Petersen, CREATION DATE: 01-Sep-79
1 !
1 ! MODIFIED BY:
1 !
1 !       DGP,01-Sep-79 : VERSION 1
1 ! 1-001 - original.  DGP 01-Sep-79
1 ! 1-002 - Remove references to OTS$$A_CUR_LUB, so this module need not be
1 !           in the sharable library.  JBS 13-SEP-1979
1 ! 1-003 - Finish development for FT2.  DGP 02-Oct-79
1 ! 1-004 - More work on MAT PRINT.  DGP 05-Oct-79
1 ! 1-005 - Work on MAT I/O for strings.  DGP 10-Oct-79
1 ! 1-006 - MAT PRINT initializing UPPER_BOUND1 incorrectly.  DGP 15-Oct-79
1 ! 1-007 - MAT INPUT, READ, and LINPUT not initializing UPPER_BOUND1 properly.
1 !         DGP 16-Oct-79
1 ! 1-008 - Bug fix in 2 dimensional MAT PRINT with both optional args.  DGP
```

```
;    58          0058  1 !           14-Nov-79
;    59          0059  1 ! 1-009 - Deallocate any temporary storage allocated.  DGP 14-Nov-79
;    60          0060  1 ! 1-010 - Move the BUILTIN ACTUALCOUNT declaration inside the routines that
;    61          0061  1 !           need it.  JBS 20-Aug-1980
;    62          0062  1 ! 1-011 - Add support for byte, g & h floating.  PLL 22-Sep-81
;    63          0063  1 ! 1-012 - Add support for decimal arrays, and dynamically mapped arrays.  PLL 23-Mar-1982
;    64          0064  1 ! 1-013 - Fix bug in MAT PRINT of strings.  Null strings caused an error.
;    65          0065  1 !           PLL 31-Mar-1982
;    66          0066  1 ! 1-014 - Pass all longwords in the calls to the g and h store routines.
;    67          0067  1 !           PLL 8-Apr-1982
;    68          0068  1 ! 1-015 - TEMP_STORE in IN_MAT should be 4 longwords.  PLL 8-Apr-1982
;    69          0069  1 ! 1-016 - TEMP_STORE [0] should be cleared out before FETCHing dynamically
;    70          0070  1 !           mapped byte or word array elements.  DG 13-Jan-1984
;    71          0071  1 !--
;    72          0072  1
;    73          0073  1 !<BLF/PAGE>
;   75          0074   1 !
;   76          0075   1 ! SWITCHES
;   77          0076   1 !
;   78          0077   1
;   79          0078   1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;   80          0079   1
;   81          0080   1 !
;   82          0081   1 ! LINKAGES
;   83          0082   1 !
;   84          0083   1
;   85          0084   1 REQUIRE 'RTLIN:BASLNK';                            ! Some Basic specific linkages
;   86          0161   1
;   87          0162   1 REQUIRE 'RTLIN:OTSLNK';                            ! All of the rest of the linkages
;   88          0591   1
;   89          0592   1 !
;   90          0593   1 ! TABLE OF CONTENTS:
;   91          0594   1 !
;   92          0595   1
;   93          0596   1 FORWARD ROUTINE
;   94          0597   1     BAS$NUM,                                       ! Returns the value of NUM
;   95          0598   1     BAS$NUM2,                                      ! returns the value of NUM2
;   96          0599   1     BAS$$NUM2_INIT : NOVALUE,                      ! initialize NUM2
;   97          0600   1     BAS$$NUM_INIT : NOVALUE,                       ! initialize NUM
;   98          0601   1     BAS$OUT_MAT_S : NOVALUE,                       ! Matrix PRINT, semicolon format
;   99          0602   1     BAS$OUT_MAT_C : NOVALUE,                       ! Matrix PRINT, comma format
;  100          0603   1     BAS$OUT_MAT_B : NOVALUE,                       ! Matrix PRINT, no format
;  101          0604   1     BAS$IN_MAT : NOVALUE;                          ! Matrix INPUT
;  102          0605   1
;  103          0606   1 !
;  104          0607   1 ! INCLUDE FILES:
;  105          0608   1 !
;  106          0609   1
;  107          0610   1 REQUIRE 'RTLML:BASPAR';                            ! some Basic constants
;  108          0632   1
;  109          0633   1 REQUIRE 'RTLIN:RTLPSECT';                          ! Psect definitions
;  110          0728   1
;  111          0729   1 REQUIRE 'RTLML:OTSISB';                            ! I/O statement block (ISB) offsets
;  112          0897   1
;  113          0898   1 REQUIRE 'RTLML:OTSLUB';                            ! Logical Unit Block (LUB) offsets
;  114          1038   1
;  115          1039   1 LIBRARY 'RTLSTARLE';                               ! system macros and symbols
;  116          1040   1
;  117          1041   1 !
;  118          1042   1 ! MACROS:
;  119          1043   1 !
;  120          1044   1
;  121          1045   1 MACRO
;  122        M 1046   1     U1_1D =
;  123          1047   1  28, 0, 32, 0%,                                    ! upper bound, one dimensional array
;  124        M 1048   1     U1_2D =
;  125          1049   1  32, 0, 32, 0%,                                    ! first upper bound, two dim. array
;  126        M 1050   1     U2_2D =
;  127          1051   1  40, 0, 32, 0%;                                    ! second upper bound, two dim. array
;  128          1052   1
;  129          1053   1 !
;  130          1054   1 ! PSECT DECLARATIONS:
;  131          1055   1 !
```

```
1 DECLARE_PSECTS (BAS);                                  ! Basic psects
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !      NONF
1 !
1 ! OWN STORAGE:
1 !
1
1 OWN
1     NUM : INITIAL (0),                                 ! Number of columns entered
1     NUM2 : INITIAL (0);                                ! Number of rows entered for 2 dim. array else 0
1
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 EXTERNAL ROUTINE
1     STR$FREE1_DX,                                      ! Free a dynamic string
1     BAS$$BLNK_LINE : CALL_CCB NOVALUE,                 ! write a blank line
1     BAS$$UDF_RL1 : CALL_CCB,                           ! UDF level - read list directed
1     BAS$$UDF_WL1 : CALL_CCB NOVALUE,                   ! UDF level - write list directed
1     BAS$FETCH_BFA : NOVALUE,                           ! Call - fetch from an array
1     BAS$STORE_BFA : NOVALUE,                           ! Call - store into an array
1     BAS$$STOP : NOVALUE,                               ! signal an error and stop
1     BAS$$CB_GET : JSB_CB_GET NOVALUE,                  ! Load CUR_LUB into register CCB
1     BAS$FET_FA_B_R8 : VA_JSB,                          ! fetch from byte array
1     BAS$FET_FA_W_R8 : VA_JSB,                          ! fetch from word array
1     BAS$FET_FA_L_R8 : VA_JSB,                          ! fetch from longword array
1     BAS$FET_FA_F_R8 : VA_JSB,                          ! fetch from floating array
1     BAS$FET_FA_D_R8 : VA_JSB,                          ! fetch from double array
1     BAS$FET_FA_G_R8 : VA_JSB,                          ! fetch from gfloat array
1     BAS$FET_FA_H_R8 : VA_JSB,                          ! fetch from hfloat array
1     BAS$STO_FA_B_R8 : VA_JSB,                          ! store into byte array
1     BAS$STO_FA_W_R8 : VA_JSB NOVALUE,                  ! store into word array
1     BAS$STO_FA_L_R8 : VA_JSB NOVALUE,                  ! store into longword array
1     BAS$STO_FA_F_R8 : VA_JSB NOVALUE,                  ! store into floating array
1     BAS$STO_FA_D_R8 : VA_JSB NOVALUE,                  ! store into double array
1     BAS$STO_FA_G_R8 : VA_JSB NOVALUE,                  ! store into gfloat array
1     BAS$STO_FA_H_R8 : VA_JSB NOVALUE,                  ! store into hfloat array
1     BAS$FETCH_DESC;                                    ! fetch elem from array of desc
1 EXTERNAL LITERAL
1     BAS$K_DATTYPERR : UNSIGNED (8);                    ! Data type error
1
```

```
1 GLOBAL ROUTINE BAS$NUM                            ! NUM
1     : =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine supports the Basic NUM function.  It returns the number of
1 !       rows input in a two dimensional array and the number of elements input
1 !       in a one dimensional array.  It uses a chunk of OWN storage because
1 !       those are the Basic semantics.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NUM.rl                  The number of elements or rows read
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NUMBER_OF_ELEMENTS.wl.v number of elements read on last MAT INPUT, LINPUT, or READ.
1 !
1 ! SIDE EFFECTS:
1 !
1 !--
1
2     BEGIN
2     RETURN .NUM
1     END;                                          ! End of BAS$NUM

                                                    .TITLE  BAS$MAT_IO
                                                    .IDENT  \1-016\

                                                    .PSECT  _BAS$DATA,NOEXE,  PIC,2

                          00000000  00000 NUM:      .LONG   0                                   ;
                          00000000  00004 NUM2:     .LONG   0                                   ;

                                                    .EXTRN  STR$FREE1_DX, BAS$$BLNK_LINE
                                                    .EXTRN  BAS$$UDF_RL1, BAS$$UDF_WL1
                                                    .EXTRN  BAS$FETCH_BFA, BAS$STORE_BFA
                                                    .EXTRN  BAS$$STOP, BAS$$CB_GET
                                                    .EXTRN  BAS$FET_FA_B_R8
                                                    .EXTRN  BAS$FET_FA_W_R8
                                                    .EXTRN  BAS$FET_FA_L_R8
                                                    .EXTRN  BAS$FET_FA_F_R8
                                                    .EXTRN  BAS$FET_FA_D_R8
                                                    .EXTRN  BAS$FET_FA_G_R8
                                                    .EXTRN  BAS$FET_FA_H_R8
                                                    .EXTRN  BAS$STO_FA_B_R8
                                                    .EXTRN  BAS$STO_FA_W_R8
```

```
                                                                       .EXTRN  BAS$STO_FA_L_R8
                                                                       .EXTRN  BAS$STO_FA_F_R8
                                                                       .EXTRN  BAS$STO_FA_D_R8
                                                                       .EXTRN  BAS$STO_FA_G_R8
                                                                       .EXTRN  BAS$STO_FA_H_R8
                                                                       .EXTRN  BAS$FETCH_DESC, BAS$K_DATTYPERR

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

                                                   0000 00000          .ENTRY  BAS$NUM, Save nothing                          ; 1100
                                    50 00000000'  EF  D0 00002         MOVL    NUM, R0                                        ; 1132
                                                    04 00009           RET                                                    ; 1133

; Routine Size:  10 bytes,    Routine Base:  _BAS$CODE + 0000


;  211          1134  1
;   213         1135  1 GLOBAL ROUTINE BAS$NUM2                              ! NUM2
;   214         1136  1     : =
;   215         1137  1
;   216         1138  1 !++
;   217         1139  1 ! FUNCTIONAL DESCRIPTION:
;   218         1140  1 !
;   219         1141  1 !       This routine supports the Basic NUM2 function.  It returns the number
;   220         1142  1 !       of elements entered in the last row of a 2 dimensional array or 0.
;   221         1143  1 !
;   222         1144  1 ! FORMAL PARAMETERS:
;   223         1145  1 !
;   224         1146  1 !       NONE
;   225         1147  1 !
;   226         1148  1 ! IMPLICIT INPUTS:
;   227         1149  1 !
;   228         1150  1 !       NUM2.rl          The number of elements read in the last row or 0
;   229         1151  1 !
;   230         1152  1 ! IMPLICIT OUTPUTS:
;   231         1153  1 !
;   232         1154  1 !       NONE
;   233         1155  1 !
;   234         1156  1 ! ROUTINE VALUE:
;   235         1157  1 !
;   236         1158  1 !       NUM_OF_ELEMENTS.wl.v    number of elements
;   237         1159  1 !
;   238         1160  1 ! SIDE EFFECTS:
;   239         1161  1 !
;   240         1162  1 !--
;   241         1163  1
;   242         1164  2     BEGIN
;   243         1165  2     RETURN .NUM2
;   244         1166  1     END;                                             ! End of BAS$NUM2



                                         0000 00000         .ENTRY  BAS$NUM2, Save nothing                            ; 1135
                         50 00000000'  EF  D0 00002         MOVL    NUM2, R0                                          ; 1165
                                           04 00009         RET                                                       ; 1166

; Routine Size:  10 bytes,    Routine Base:  _BAS$CODE + 000A


;   245         1167  1
;  247           1168  1 GLOBAL ROUTINE BAS$$NUM_INIT                              ! NUM_INIT
;  248           1169  1     : NOVALUE =
;  249           1170  1
;  250           1171  1 !++
;  251           1172  1 ! FUNCTIONAL DESCRIPTION:
;  252           1173  1 !
;  253           1174  1 !       Initialize NUM to 0.
;  254           1175  1 !
;  255           1176  1 ! FORMAL PARAMETERS:
;  256           1177  1 !
;  257           1178  1 !       NONE
;  258           1179  1 !
;  259           1180  1 ! IMPLICIT INPUTS:
;  260           1181  1 !
;  261           1182  1 !       NONE
;  262           1183  1 !
;  263           1184  1 ! IMPLICIT OUTPUTS:
;  264           1185  1 !
;  265           1186  1 !       NUM.wl                   The number of rows read
;  266           1187  1 !
;  267           1188  1 ! ROUTINE VALUE:
;  268           1189  1 !
;  269           1190  1 !       NONE
;  270           1191  1 !
;  271           1192  1 ! SIDE EFFECTS:
;  272           1193  1 !
;  273           1194  1 !--
;  274           1195  1
;  275           1196  2     BEGIN
;  276           1197  2     NUM = 0;
;  277           1198  2     RETURN;
;  278           1199  1     END;                                          ! End of BAS$$NUM_INIT



                                             0000 00000          .ENTRY  BAS$$NUM_INIT, Save nothing                    ; 1168
                                00000000'  EF  D4 00002          CLRL    NUM                                            ; 1197
                                               04 00008          RET                                                    ; 1199

; Routine Size:  9 bytes,    Routine Base:  _BAS$CODE + 0014


;  279           1200  1
;  281        1201  1 GLOBAL ROUTINE BAS$$NUM2_INIT                         ! Initialize NUM2
;  282        1202  1     : NOVALUE =
;  283        1203  1
;  284        1204  1 !++
;  285        1205  1 ! FUNCTIONAL DESCRIPTION:
;  286        1206  1 !
;  287        1207  1 !       This routine initializes NUM2 to 0.
;  288        1208  1 !
;  289        1209  1 ! FORMAL PARAMETERS:
;  290        1210  1 !
;  291        1211  1 !       NONE
;  292        1212  1 !
;  293        1213  1 ! IMPLICIT INPUTS:
;  294        1214  1 !
;  295        1215  1 !       NONE
;  296        1216  1 !
;  297        1217  1 ! IMPLICIT OUTPUTS:
;  298        1218  1 !
;  299        1219  1 !       NUM2.wl              Number of columns in last row.
;  300        1220  1 !
;  301        1221  1 ! ROUTINE VALUE:
;  302        1222  1 !
;  303        1223  1 !       NONE
;  304        1224  1 !
;  305        1225  1 ! SIDE EFFECTS:
;  306        1226  1 !
;  307        1227  1 !--
;  308        1228  1
;  309        1229  2     BEGIN
;  310        1230  2     NUM2 = 0;
;  311        1231  2     RETURN;
;  312        1232  1     END;                                            ! End of BAS$$NUM2_INIT


                                       0000 00000        .ENTRY  BAS$$NUM2_INIT, Save nothing                ; 1201
                          00000000'  EF  D4 00002        CLRL    NUM2                                        ; 1230
                                         04 00008        RET                                                 ; 1232

; Routine Size:  9 bytes,    Routine Base:  _BAS$CODE + 001D


;  313        1233  1
```

```
1 GLOBAL ROUTINE BAS$OUT_MAT_S (                        ! Matrix print, semicolon format
1         ARRAY, SUBSCRIPT1, SUBSCRIPT2) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       The array is printed one element at a time with the elements in each row
1 !       being printed in a packed format.  Each row begins on a new line.  Row
1 !       and column zero are not printed.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       ARRAY.rx.a                        ! array to print
1 !       [SUBSCRIPT1.rlu.v]                ! first optional subscript
1 !       [SUBSCRIPT2.rlu.v]                ! second optional subscript
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Signals:
1 !       Data Type Error
1 !
1 !--
1
2   BEGIN
2
2   GLOBAL REGISTER
2       CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2   BUILTIN
2       ACTUALCOUNT;
2
2   LITERAL
2       V_1D_FLAG = 1,                          ! flag - one dimen. array
2       K_ONE_OPT_ARG = 2,                      ! value of arg. count for one
2                                               ! optional argument
2       K_TWO_OPT_ARGS = 3,                     ! value of arg. count for two
2                                               ! optional arguments
2       K_1D = 1;                               ! one dimension
2
2   LOCAL
2       NUM_ELEMS_DONE,                         ! total number of array elements processed
2       FLAGS,
2       TEMP_STORE : VECTOR [4, LONG],          ! temp storage for calling FETCH_VA
2       ROW,                                    ! current value of subscript 1
2       COLUMN,                                 ! current value of subscript 2
```

```
;  372        1291  2         UPPER_BOUND1,                          ! upper bound for 1 dimensional
;  373        1292  2                                                ! array and number of rows for 2
;  374        1293  2                                                ! dimensional array
;  375        1294  2         TOTAL_NUM_ITEMS,                       ! total number of items in the array
;  376        1295  2                                                ! excluding row and col. 0
;  377        1296  2         ELEM_DESCRIP : REF BLOCK [12,BYTE],    ! desc fetched from array
;  378        1297  2         NUM_DESCRIP : BLOCK [8,BYTE];          ! numeric desc for FETCH
;  379        1298  2
;  380        1299  2     MAP
;  381        1300  2         ARRAY : REF BLOCK [, BYTE];
;  382        1301  2
;  383        1302  2     BAS$$CB_GET ();
;  384        1303  2  !+
;  385        1304  2  ! Check to see if this a list of arrays.  If it is, then print a blank line between
;  386        1305  2  ! each array.
;  387        1306  2  !-
;  388        1307  2
;  389        1308  2     IF .CCB [ISB$V_MAT_PRINT] THEN BAS$$BLNK_LINE ();
;  390        1309  2
;  391        1310  2     CCB [ISB$V_MAT_PRINT] = 1;
;  392        1311  2     FLAGS = 0;
;  393        1312  2  !+
;  394        1313  2  ! Default TEMP_STORE to a dynamic stirng descriptor
;  395        1314  2  !-
;  396        1315  2     TEMP_STORE [0] = %X'020E0000';
;  397        1316  2     TEMP_STORE [1] = %X'00000000';
;  398        1317  2  !+
;  399        1318  2  ! Check the number of dimensions and set a flag if only one dimension.
;  400        1319  2  !-
;  401        1320  2
;  402        1321  2     IF .ARRAY [DSC$B_DIMCT] EQL K_1D THEN FLAGS = .FLAGS + V_1D_FLAG;
;  403        1322  2
;  404        1323  2  !+
;  405        1324  2  ! Check for optional arguments.  If there are no optional arguments, then set
;  406        1325  2  ! the upper bounds based on what is in the descriptor.  If there are optional
;  407        1326  2  ! args, then use them as the upper bound.
;  408        1327  2  !-
;  409        1328  2
;  410        1329  2     IF ACTUALCOUNT () LSS K_ONE_OPT_ARG
;  411        1330  2     THEN
;  412        1331  2
;  413        1332  2         IF .ARRAY [DSC$B_DIMCT] EQL K_1D
;  414        1333  2         THEN
;  415        1334  2  !+
;  416        1335  2  ! No optional arguments and a one dimensional array
;  417        1336  2  !-
;  418        1337  3             BEGIN
;  419        1338  3             UPPER_BOUND1 = .ARRAY [U1_1D];
;  420        1339  3             TOTAL_NUM_ITEMS = .UPPER_BOUND1;
;  421        1340  3             END
;  422        1341  2         ELSE
;  423        1342  3             BEGIN
;  424        1343  3  !+
;  425        1344  3  ! 2 dimensional array
;  426        1345  3  !-
;  427        1346  3             UPPER_BOUND1 = .ARRAY [U2_2D];
;  428        1347  3             TOTAL_NUM_ITEMS = .ARRAY [U1_2D]*.UPPER_BOUND1;
```

```
2                 END;
2
2         IF ACTUALCOUNT () GEQ K_ONE_OPT_ARG
2         THEN
3             BEGIN
3             UPPER_BOUND1 = .SUBSCRIPT1;
3             TOTAL_NUM_ITEMS = .SUBSCRIPT1;
2             END;
2
2         IF ACTUALCOUNT () EQL K_TWO_OPT_ARGS
2         THEN
2 !+
2 ! 2 optional arguments
2 !-
3             BEGIN
3             UPPER_BOUND1 = .SUBSCRIPT2;
3             TOTAL_NUM_ITEMS = .SUBSCRIPT1*.SUBSCRIPT2;
2             END;
2
2 !+
2 ! Initialize the two current subscripts regardless of the number of dimensions
2 !-
2         ROW = COLUMN = NUM_ELEMS_DONE = 1;
2 !+
2 ! Check for array of descriptors.  They could be dynamic string descriptors,
2 ! or numeric descriptors for a dynamically mapped array.  Fetch
2 ! an element (a descriptor) from the array and check the dtype to
2 ! determine if this is a string array or numeric array.
2 !-
2
2         IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC
2         THEN
3         BEGIN
3
3         NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
3         IF .FLAGS AND V_1D_FLAG
3         THEN
3             ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1)
3         ELSE
3             ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1, 1);
3
3
3         CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
3             SET
3
3             [DSC$K_DTYPE_T] :               ! text
3
3                 ;
3
3             [DSC$K_DTYPE_B] :               ! byte
3
4                 BEGIN
4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_B;
4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/4;
3                 END;
3
```

```
[DSC$K_DTYPE_W] :                   . word

    BEGIN
    NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
    NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_W;
    NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/2;
    END;

[DSC$K_DTYPE_L] :                   ! long

    BEGIN
    NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
    NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_L;
    NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
    END;

[DSC$K_DTYPE_F] :                   ! float

    BEGIN
    NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
    NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_F;
    NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
    END;

[DSC$K_DTYPE_D] :                   ! double

    BEGIN
    NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
    NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_D;
    NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
    END;

[DSC$K_DTYPE_G] :                   ! g float

    BEGIN
    NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
    NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_G;
    NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
    END;

[DSC$K_DTYPE_H] :                   ! h float

    BEGIN
    NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
    NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_H;
    NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*4;
    END;

[DSC$K_DTYPE_P] :                   ! packed decimal

    BEGIN
    NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
    NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
    NUM_DESCRIP [DSC$W_LENGTH] = .ARRAY [DSC$W_LENGTH];
    NUM_DESCRIP [DSC$B_SCALE] = .ARRAY [DSC$B_SCALE];
    END;
```

```
;  3          [INRANGE,OUTRANGE] :
;  3              BAS$$STOP (BAS$K_DATTYPERR);
;  3
;  3          TES;
;  3
;  2      END;
;  2  !+
;  2  ! Loop thru the array descriptor until all of the elements in the array or as
;  2  ! specified by the optional arguments have been printed.  Start each row on a
;  2  ! new line.
;  2  !-
;  2
;  2      WHILE .NUM_ELEMS_DONE LEQ .TOTAL_NUM_ITEMS DO
;  3          BEGIN
;  3  !+
;  3  ! Based on the data type, JSB or CALL the proper fetch routine to get the element
;  3  ! out of the array.  The FETCH and STORE routines are called because the array
;  3  ! may be virtual.
;  3  !-
;  3
;  3          CASE .ARRAY [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  3              SET
;  3
;  3              [DSC$K_DTYPE_B] :
;  3
;  3                  IF .FLAGS AND V_1D_FLAG
;  3                  THEN
;  3                      TEMP_STORE [0] = BAS$FET_FA_B_R8 (.ARRAY, .COLUMN)
;  3                  ELSE
;  3                      TEMP_STORE [0] = BAS$FET_FA_B_R8 (.ARRAY, .ROW, .COLUMN);
;  3
;  3              [DSC$K_DTYPE_W] :
;  3
;  3                  IF .FLAGS AND V_1D_FLAG
;  3                  THEN
;  3                      TEMP_STORE [0] = BAS$FET_FA_W_R8 (.ARRAY, .COLUMN)
;  3                  ELSE
;  3                      TEMP_STORE [0] = BAS$FET_FA_W_R8 (.ARRAY, .ROW, .COLUMN);
;  3
;  3              [DSC$K_DTYPE_L] :
;  3
;  3                  IF .FLAGS AND V_1D_FLAG
;  3                  THEN
;  3                      TEMP_STORE [0] = BAS$FET_FA_L_R8 (.ARRAY, .COLUMN)
;  3                  ELSE
;  3                      TEMP_STORE [0] = BAS$FET_FA_L_R8 (.ARRAY, .ROW, .COLUMN);
;  3
;  3              [DSC$K_DTYPE_F] :
;  3
;  3                  IF .FLAGS AND V_1D_FLAG
;  3                  THEN
;  3                      TEMP_STORE [0] = BAS$FET_FA_F_R8 (.ARRAY, .COLUMN)
;  3                  ELSE
;  3                      TEMP_STORE [0] = BAS$FET_FA_F_R8 (.ARRAY, .ROW, .COLUMN);
;  3
;  3              [DSC$K_DTYPE_D] :
;  3
```

```
:  600        1519  3              IF .FLAGS AND V_1D_FLAG
:  601        1520  3              THEN
:  602        1521  3                  TEMP_STORE [0] = BAS$FET_FA_D_R8 (.ARRAY, .COLUMN)
:  603        1522  3              ELSE
:  604        1523  3                  TEMP_STORE [0] = BAS$FET_FA_D_R8 (.ARRAY, .ROW, .COLUMN);
:  605        1524  3
:  606        1525  3          [DSC$K_DTYPE_T] :
:  607        1526  3
:  608        1527  3              IF .FLAGS AND V_1D_FLAG
:  609        1528  3              THEN
:  610        1529  3                  BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .COLUMN)
:  611        1530  3              ELSE
:  612        1531  3                  BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .ROW, .COLUMN);
:  613        1532  3
:  614        1533  3          [DSC$K_DTYPE_DSC] :
:  615        1534  3
:  616        1535  4              BEGIN
:  617        1536  4              CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
:  618        1537  4                  SET
:  619        1538  4
:  620        1539  4                  [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F,
:  621        1540  4                   DSC$K_DTYPE_D, DSC$K_DTYPE_G, DSC$K_DTYPE_H, DSC$K_DTYPE_P] :
:  622        1541  5                      BEGIN
:  623        1542  5
:  624        1543  5                      TEMP_STORE [0] = %X'00000000';
:  625        1544  5                      IF .FLAGS AND V_1D_FLAG
:  626        1545  5                      THEN
:  627        1546  5                          BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .COLUMN)
:  628        1547  5                      ELSE
:  629        1548  5                          BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .ROW, .COLUMN);
:  630        1549  5
:  631        1550  4                      END;
:  632        1551  4
:  633        1552  4                  [DSC$K_DTYPE_T] :
:  634        1553  4
:  635        1554  4                      IF .FLAGS AND V_1D_FLAG
:  636        1555  4                      THEN
:  637        1556  4                          BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .COLUMN)
:  638        1557  4                      ELSE
:  639        1558  4                          BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .ROW, .COLUMN);
:  640        1559  4
:  641        1560  4                  [INRANGE,OUTRANGE] :
:  642        1561  4                      BAS$$STOP (BAS$K_DATTYPERR);
:  643        1562  4
:  644        1563  4                  TES;
:  645        1564  4
:  646        1565  3              END;                                ! end of dtype dsc
:  647        1566  3
:  648        1567  3          [DSC$K_DTYPE_P] :
:  649        1568  3          !+
:  650        1569  3          ! Must pass a descriptor to BAS$$UDF_WL1.  Construct a class SD
:  651        1570  3          ! descriptor here, and set the pointer field to TEMP_STORE.
:  652        1571  3          !-
:  653        1572  4              BEGIN
:  654        1573  4              NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
:  655        1574  4              NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
:  656        1575  4              NUM_DESCRIP [DSC$W_LENGTH] = .ARRAY [DSC$W_LENGTH];
```

```
;  657        1576  4             NUM_DESCRIP [DSC$B_SCALE] = .ARRAY [DSC$B_SCALE];
;  658        1577  4             NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
;  659        1578  4
;  660        1579  4             IF .FLAGS AND V_1D_FLAG
;  661        1580  4             THEN
;  662        1581  4                 BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .COLUMN)
;  663        1582  4             ELSE
;  664        1583  4                 BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .ROW, .COLUMN);
;  665        1584  3             END;
;  666        1585  3
;  667        1586  3         [DSC$K_DTYPE_G] :
;  668        1587  3             IF .FLAGS AND V_1D_FLAG
;  669        1588  3             THEN
;  670        1589  3                 TEMP_STORE [0] = BAS$FET_FA_G_R8 (.ARRAY, .COLUMN)
;  671        1590  3             ELSE
;  672        1591  3                 TEMP_STORE [0] = BAS$FET_FA_G_R8 (.ARRAY, .ROW, .COLUMN);
;  673        1592  3
;  674        1593  3         [DSC$K_DTYPE_H] :
;  675        1594  3
;  676        1595  3             IF .FLAGS AND V_1D_FLAG
;  677        1596  3             THEN
;  678        1597  3                 TEMP_STORE [0] = BAS$FET_FA_H_R8 (.ARRAY, .COLUMN)
;  679        1598  3             ELSE
;  680        1599  3                 TEMP_STORE [0] = BAS$FET_FA_H_R8 (.ARRAY, .ROW, .COLUMN);
;  681        1600  3
;  682        1601  3         [INRANGE, OUTRANGE] :
;  683        1602  3             BAS$$STOP (BAS$K_DATTYPERR);
;  684        1603  3         TES;
;  685        1604  3
;  686        1605  3     BAS$$UDF_WL1 (
;  687        1606  4         BEGIN
;  688        1607  4
;  689        1608  4         IF (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC) THEN .ELEM_DESCRIP [DSC$B_DTYPE] ELSE .ARRAY [DSC$
;  690        1609  4
;  691        1610  4         END
;  692        1611  3     ,                                       !
;  693        1612  4         BEGIN
;  694        1613  4         MAP
;  695        1614  4             TEMP_STORE : BLOCK [8,BYTE];
;  696        1615  4
;  697        1616  5         (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T
;  698        1617  5          THEN
;  699        1618  5             .TEMP_STORE [DSC$W_LENGTH]
;  700        1619  5          ELSE
;  701        1620  6             (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC
;  702        1621  6              THEN
;  703        1622  6                 IF .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T
;  704        1623  6                 THEN
;  705        1624  6                     .TEMP_STORE [DSC$W_LENGTH]
;  706        1625  6                 ELSE
;  707        1626  6                     .NUM_DESCRIP [DSC$W_LENGTH]
;  708        1627  6              ELSE
;  709        1628  5                 .ARRAY [DSC$W_LENGTH]))
;  710        1629  4         END
;  711        1630  3     ,                                       !
;  712        1631  4         (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  713        1632  4          THEN
;   714        1633 4                 NUM_DESCRIP                         ! pass dsc for packed
;   715        1634 4              ELSE
;   716        1635 5                 (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
;   717        1636 5                     .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;   718        1637 5                  THEN
;   719        1638 5                     NUM_DESCRIP
;   720        1639 5                  ELSE
;   721        1640 3                     TEMP_STORE)),
;   722        1641 3         ! If this is the last element of this row, then pass the "no format"
;   723        1642 3         ! argument so that the first element of the next row starts on a
;   724        1643 3         ! new line
;   725        1644 4             BEGIN
;   726        1645 4
;   727        1646 4             IF (.COLUMN EQL .UPPER_BOUND1) THEN BAS$K_NO_FORM ELSE BAS$K_SEMI_FORM
;   728        1647 4
;   729        1648 4             END
;   730        1649 3         );
;   731        1650 3         NUM_ELEMS_DONE = .NUM_ELEMS_DONE + 1;
;   732        1651 3         COLUMN = .COLUMN + 1;
;   733        1652 3
;   734        1653 3         IF .COLUMN GTR .UPPER_BOUND1
;   735        1654 3         THEN
;   736        1655 4             BEGIN
;   737        1656 4 !+
;   738        1657 4 ! It is time to start a new row.
;   739        1658 4 !-
;   740        1659 4             ROW = .ROW + 1;
;   741        1660 4             COLUMN = 1;
;   742        1661 3             END;
;   743        1662 3
;   744        1663 2         END;                                        ! end of the WHILE loop
;   745        1664 2
;   746        1665 2 !+
;   747        1666 2 ! Return any temporary storage used and then return
;   748        1667 2 !-
;   749        1668 2
;   750        1669 2     IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T OR
;   751        1670 3       (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
;   752        1671 3        .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T)
;   753        1672 2     THEN
;   754        1673 2         STR$FREE1_DX (TEMP_STORE);
;   755        1674 2
;   756        1675 2     RETURN;
;   757        1676 1     END;                                        !End of BAS$OUT_MAT_S


                                     0FFC 00000          .ENTRY  BAS$OUT_MAT_S, Save R2,R3,R4,R5,R6,R7,R8,-  ; 1234
                                                                 R9,R10,R11                                  ;
                     5E              30  C2 00002          SUBL2   #48, SP                                     ;
                          00000000G  00  16 00005          JSB     BAS$$CB_GET                                 ; 1302
      07         97  AB              02  E1 0000B          BBC     #2, -105(CCB), 1$                           ; 1308
        00000000G    00              00  FB 00010          CALLS   #0, BAS$$BLNK_LINE                          ;
                 97  AB              04  88 00017 1$:      BISB2   #4, -105(CCB)                               ; 1310
                          08     AE  D4 0001B              CLRL    FLAGS                                       ; 1311
```

```
                      20  AE 020E0000  8F  D0 0001E        MOVL    #34471936, TEMP_STORE                       ; 1315
                                  24   AE  D4 00026        CLRL    TEMP_STORE+4                                ; 1316
                          59      04   AC  D0 00029        MOVL    ARRAY, R9                                   ; 1321
                                       50  D4 0002D        CLRL    R0                                          ;
                          01      0B   A9  91 0002F        CMPB    11(R9), #1                                  ;
                                       05  12 00033        BNEQ    2$                                          ;
                                       50  D6 00035        INCL    R0                                          ;
                                  08   AE  D6 00037        INCL    FLAGS                                       ;
                          02           6C  91 0003A 2$:    CMPB    (AP), #2                                    ; 1329
                                       1B  1E 0003D        BGEQU   4$                                          ;
                          0C           50  E9 0003F        BLBC    R0, 3$                                      ; 1332
                      0C  AE      1C   A9  D0 00042        MOVL    28(R9), UPPER_BOUND1                        ; 1338
                      10  AE      0C   AE  D0 00047        MOVL    UPPER_BOUND1, TOTAL_NUM_ITEMS               ; 1339
                                       0C  11 0004C        BRB     4$                                          ; 1332
                      0C  AE      28   A9  D0 0004E 3$:    MOVL    40(R9), UPPER_BOUND1                        ; 1346
          10    AE    20  A9      0C   AE  C5 00053        MULL3   UPPER_BOUND1, 32(R9), TOTAL_NUM_ITEMS       ; 1347
                          02           6C  91 0005A 4$:    CMPB    (AP), #2                                    ; 1350
                                       0A  1F 0005D        BLSSU   5$                                          ;
                      0C  AE      08   AC  D0 0005F        MOVL    SUBSCRIPT1, UPPER_BOUND1                    ; 1353
                      10  AE      08   AC  D0 00064        MOVL    SUBSCRIPT1, TOTAL_NUM_ITEMS                 ; 1354
                          03           6C  91 00069 5$:    CMPB    (AP), #3                                    ; 1357
                                       0C  12 0006C        BNEQ    6$                                          ;
                      0C  AE      0C   AC  D0 0006E        MOVL    SUBSCRIPT2, UPPER_BOUND1                    ; 1363
          10    AE    08  AC      0C   AC  C5 00073        MULL3   SUBSCRIPT2, SUBSCRIPT1, TOTAL_NUM_ITEMS     ; 1364
                      14  AE           01  D0 0007A 6$:    MOVL    #1, NUM_ELEMS_DONE                          ; 1370
                          6E           01  D0 0007E        MOVL    #1, COLUMN                                  ;
                      04  AE           01  D0 00081        MOVL    #1, ROW                                     ;
                          18      02   A9  91 00085        CMPB    2(R9), #24                                  ; 1378
                                       78  12 00089        BNEQ    13$                                         ;
                      1C  AE      20   AE  9E 0008B        MOVAB   TEMP_STORE, NUM_DESCRIP+4                   ; 1382
                      0D  AE      08   AE  E9 00090        BLBC    FLAGS, 7$                                   ; 1383
                                       01  DD 00094        PUSHL   #1                                          ; 1385
                                       59  DD 00096        PUSHL   R9                                          ;
               00000000G  00           02  FB 00098        CALLS   #2, BAS$FETCH_DESC                          ;
                                       0D  11 0009F        BRB     8$                                          ;
                                       01  DD 000A1 7$:    PUSHL   #1                                          ; 1387
                                       01  DD 000A3        PUSHL   #1                                          ;
                                       59  DD 000A5        PUSHL   R9                                          ;
               00000000G  00           03  FB 000A7        CALLS   #3, BAS$FETCH_DESC                          ;
                          5A           50  D0 000AE 8$:    MOVL    R0, ELEM_DESCRIP                            ;
                16        06      02   AA  8F 000B1        CASEB   2(ELEM_DESCRIP), #6, #22                    ; 1390
   002E       004F      0045         003B     000B6 9$:    .WORD   11$-9$,-                                    ;
   002E       002E      0063         0057     000BE                12$-9$,-                                    ;
   002E       002E      002E         0090     000C6                14$-9$,-                                    ;
   0081       002E      002E         002E     000CE                10$-9$,-                                    ;
   002E       002E      002E         002E     000D6                15$-9$,-                                    ;
              0077      006B         002E     000DE                17$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   22$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   10$-9$,-                                    ;
                                                                   21$-9$,-                                    ;
```

```
                                                                         10$-9$,-                          ;
                                                                         10$-9$,-                          ;
                                                                         10$-9$,-                          ;
                                                                         10$-9$,-                          ;
                                                                         10$-9$,-                          ;
                                                                         18$-9$,-                          ;
                                                                         20$-9$                            ;
                         7E         00G  8F  9A 000E4 10$:     MOVZBL   #BAS$K_DATTYPERR, -(SP)            ; 1463
              00000000G  00              01  FB 000E8          CALLS    #1, BAS$$STOP                      ;
                                         55  11 000EF          BRB      22$                                ;
                    18   AE  01060001    8F  D0 000F1 11$:     MOVL     #17170433, NUM_DESCRIP             ; 1402
                                         4B  11 000F9          BRB      22$                                ; 1390
                    18   AE  01070002    8F  D0 000FB 12$:     MOVL     #17235970, NUM_DESCRIP             ; 1410
                                         41  11 00103 13$:     BRB      22$                                ; 1390
                    1A   AE      0108    8F  B0 00105 14$:     MOVW     #264, NUM_DESCRIP+2                ; 1417
                                         06  11 0010B          BRB      16$                                ; 1418
                    1A   AE      010A    8F  B0 0010D 15$:     MOVW     #266, NUM_DESCRIP+2                ; 1425
                    18   AE              04  B0 00113 16$:     MOVW     #4, NUM_DESCRIP                    ; 1426
                                         2D  11 00117          BRB      22$                                ; 1390
                    1A   AE      010B    8F  B0 00119 17$:     MOVW     #267, NUM_DESCRIP+2                ; 1433
                                         06  11 0011F          BRB      19$                                ; 1434
                    1A   AE      011B    8F  B0 00121 18$:     MOVW     #283, NUM_DESCRIP+2                ; 1441
                    18   AE              08  B0 00127 19$:     MOVW     #8, NUM_DESCRIP                    ; 1442
                                         19  11 0012B          BRB      22$                                ; 1390
                    18   AE  011C0010    8F  D0 0012D 20$:     MOVL     #18612240, NUM_DESCRIP             ; 1450
                                         0F  11 00135          BRB      22$                                ; 1390
                    1A   AE      0915    8F  B0 00137 21$:     MOVW     #2325, NUM_DESCRIP+2               ; 1457
                    18   AE              69  B0 0013D          MOVW     (R9), NUM_DESCRIP                  ; 1458
                    20   AE        08    A9  90 00141          MOVB     8(R9), NUM_DESCRIP+8               ; 1459
                    10   AE        14    AE  D1 00146 22$:     CMPL     NUM_ELEMS_DONE, TOTAL_NUM_ITEMS    ; 1474
                                         03  15 0014B          BLEQ     23$                                ;
                                       0206  31 0014D          BRW      74$                                ;
              16             06    02    A9  8F 00150 23$:     CASEB    2(R9), #6, #22                     ; 1482
   00F7     0067      004C             0031     00155 24$:     .WORD    25$-24$,-                          ;
   00F7     00F7      009D             0082     0015D                   28$-24$,-                          ;
   00F7     00F7      00F7             00B9     00165                   31$-24$,-                          ;
   0113     00F7      00F7             00F7     0016D                   44$-24$,-                          ;
   00F7     00C4      00F7             00F7     00175                   34$-24$,-                          ;
            0169      014E             00F7     0017D                   37$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        41$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        47$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        42$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        44$-24$,-                          ;
                                                                        53$-24$,-                          ;
                                                                        56$-24$                            ;
                                       00C6  31 00183          BRW      44$                                ; 1602
```

```
                          05          08    AE  E9 00186 25$:   BLBC    FLAGS, 26$                       ; 1487
                          51                6E  D0 0018A        MOVL    COLUMN, R1                       ; 1489
                                            07  11 0018D        BRB     27$                              ;
                          52                6E  D0 0018F 26$:   MOVL    COLUMN, R2                       ; 1491
                          51          04    AE  D0 00192        MOVL    ROW, R1                          ;
                          50                59  D0 00196 27$:   MOVL    R9, R0                           ;
                               00000000G    00  16 00199        JSB     BAS$FET_FA_B_R8                  ;
                                            6A  11 0019F        BRB     40$                              ;
                          05          08    AE  E9 001A1 28$:   BLBC    FLAGS, 29$                       ; 1495
                          51                6E  D0 001A5        MOVL    COLUMN, R1                       ; 1497
                                            07  11 001A8        BRB     30$                              ;
                          52                6E  D0 001AA 29$:   MOVL    COLUMN, R2                       ; 1499
                          51          04    AE  D0 001AD        MOVL    ROW, R1                          ;
                          50                59  D0 001B1 30$:   MOVL    R9, R0                           ;
                               00000000G    00  16 001B4        JSB     BAS$FET_FA_W_R8                  ;
                                            4F  11 001BA        BRB     40$                              ;
                          05          08    AE  E9 001BC 31$:   BLBC    FLAGS, 32$                       ; 1503
                          51                6E  D0 001C0        MOVL    COLUMN, R1                       ; 1505
                                            07  11 001C3        BRB     33$                              ;
                          52                6E  D0 001C5 32$:   MOVL    COLUMN, R2                       ; 1507
                          51          04    AE  D0 001C8        MOVL    ROW, R1                          ;
                          50                59  D0 001CC 33$:   MOVL    R9, R0                           ;
                               00000000G    00  16 001CF        JSB     BAS$FET_FA_L_R8                  ;
                                            34  11 001D5        BRB     40$                              ;
                          05          08    AE  E9 001D7 34$:   BLBC    FLAGS, 35$                       ; 1511
                          51                6E  D0 001DB        MOVL    COLUMN, R1                       ; 1513
                                            07  11 001DE        BRB     36$                              ;
                          52                6E  D0 001E0 35$:   MOVL    COLUMN, R2                       ; 1515
                          51          04    AE  D0 001E3        MOVL    ROW, R1                          ;
                          50                59  D0 001E7 36$:   MOVL    R9, R0                           ;
                               00000000G    00  16 001EA        JSB     BAS$FET_FA_F_R8                  ;
                                            19  11 001F0        BRB     40$                              ;
                          05          08    AE  E9 001F2 37$:   BLBC    FLAGS, 38$                       ; 1519
                          51                6E  D0 001F6        MOVL    COLUMN, R1                       ; 1521
                                            07  11 001F9        BRB     39$                              ;
                          52                6E  D0 001FB 38$:   MOVL    COLUMN, R2                       ; 1523
                          51          04    AE  D0 001FE        MOVL    ROW, R1                          ;
                          50                59  D0 00202 39$:   MOVL    R9, R0                           ;
                               00000000G    00  16 00205        JSB     BAS$FET_FA_D_R8                  ;
                                          00C9  31 0020B 40$:   BRW     59$                              ;
                          4C          08    AE  E9 0020E 41$:   BLBC    FLAGS, 46$                       ; 1527
                                            6E  DD 00212        PUSHL   COLUMN                           ; 1529
                                      24    AE  9F 00214        PUSHAB  TEMP_STORE                       ;
                                            6C  11 00217        BRB     49$                              ;
                16        06          02    AA  8F 00219 42$:   CASEB   2(ELEM_DESCRIP), #6, #22         ; 1536
      002E      003B      003B            003B     0021E 43$:   .WORD   45$-43$,-                        ;
      002E      002E      003B            003B     00226                45$-43$,-                        ;
      002E      002E      002E            FFF0     0022E                45$-43$,-                        ;
      003B      002E      002E            002E     00236                44$-43$,-                        ;
      002E      002E      002E            002E     0023E                45$-43$,-                        ;
                003B      003B            002E     00246                45$-43$,-                        ;
                                                                        44$-43$,-                        ;
                                                                        44$-43$,-                        ;
                                                                        41$-43$,-                        ;
                                                                        44$-43$,-                        ;
                                                                        44$-43$,-                        ;
                                                                        44$-43$,-                        ;
```

```
                                                           44$-43$,-                          ;
                                                           44$-43$,-                          ;
                                                           44$-43$,-                          ;
                                                           45$-43$,-                          ;
                                                           44$-43$,-                          ;
                                                           44$-43$,-                          ;
                                                           44$-43$,-                          ;
                                                           44$-43$,-                          ;
                                                           44$-43$,-                          ;
                                                           45$-43$,-                          ;
                                                           45$-43$                            ;
           7E       00G  8F   9A 0024C 44$:    MOVZBL  #BAS$K_DATTYPERR, -(SP)               ; 1561
 00000000G  00            01   FB 00250         CALLS   #1, BAS$$STOP                         ;
                          48   11 00257         BRB     52$                                   ;
                  20      AE   D4 00259 45$:    CLRL    TEMP_STORE                            ; 1543
                          1E   11 0025C         BRB     48$                                   ; 1544
                          6E   DD 0025E 46$:    PUSHL   COLUMN                                ; 1558
                  08      AE   DD 00260         PUSHL   ROW                                   ;
                  28      AE   9F 00263         PUSHAB  TEMP_STORE                            ;
                          30   11 00266         BRB     51$                                   ;
      1A    AE     0915   8F   B0 00268 47$:    MOVW    #2325, NUM_DESCRIP+2                  ; 1574
      18    AE            69   B0 0026E         MOVW    (R9), NUM_DESCRIP                     ; 1575
      20    AE     08     A9   90 00272         MOVB    8(R9), NUM_DESCRIP+8                  ; 1576
      1C    AE     20     AE   9E 00277         MOVAB   TEMP_STORE, NUM_DESCRIP+4             ; 1577
            10     08     AE   E9 0027C 48$:    BLBC    FLAGS, 50$                            ; 1579
                          6E   DD 00280         PUSHL   COLUMN                                ; 1581
                  1C      AE   9F 00282         PUSHAB  NUM_DESCRIP                           ;
                          59   DD 00285 49$:    PUSHL   R9                                    ;
 00000000G  00            03   FB 00287         CALLS   #3, BAS$FETCH_BFA                     ;
                          4B   11 0028E         BRB     60$                                   ;
                          6E   DD 00290 50$:    PUSHL   COLUMN                                ; 1583
                  08      AE   DD 00292         PUSHL   ROW                                   ;
                  20      AE   9F 00295         PUSHAB  NUM_DESCRIP                           ;
                          59   DD 00298 51$:    PUSHL   R9                                    ;
 00000000G  00            04   FB 0029A         CALLS   #4, BAS$FETCH_BFA                     ;
                          38   11 002A1 52$:    BRB     60$                                   ; 1482
            05     08     AE   E9 002A3 53$:    BLBC    FLAGS, 54$                            ; 1587
            51            6E   D0 002A7         MOVL    COLUMN, R1                            ; 1589
                          07   11 002AA         BRB     55$                                   ;
            52            6E   D0 002AC 54$:    MOVL    COLUMN, R2                            ; 1591
            51     04     AE   D0 002AF         MOVL    ROW, R1                               ;
            50            59   D0 002B3 55$:    MOVL    R9, R0                                ;
              00000000G   00   16 002B6         JSB     BAS$FET_FA_G_R8                       ;
                          19   11 002BC         BRB     59$                                   ;
            05     08     AE   E9 002BE 56$:    BLBC    FLAGS, 57$                            ; 1595
            51            6E   D0 002C2         MOVL    COLUMN, R1                            ; 1597
                          07   11 002C5         BRB     58$                                   ;
            52            6E   D0 002C7 57$:    MOVL    COLUMN, R2                            ; 1599
            51     04     AE   D0 002CA         MOVL    ROW, R1                               ;
            50            59   D0 002CE 58$:    MOVL    R9, R0                                ;
              00000000G   00   16 002D1         JSB     BAS$FET_FA_H_R8                       ;
      20    AE            50   D0 002D7 59$:    MOVL    R0, TEMP_STORE                        ;
      0C    AE            6E   D1 002DB 60$:    CMPL    COLUMN, UPPER_BOUND1                  ; 1646
                          04   12 002DF         BNEQ    61$                                   ;
                          03   DD 002E1         PUSHL   #3                                    ;
                          02   11 002E3         BRB     62$                                   ;
                          01   DD 002E5 61$:    PUSHL   #1                                    ;
```

```
                        50      02   A9  9A 002E7 62$:    MOVZBL  2(R9), R0                          ; 1631
                        15           50  91 002EB         CMPB    R0, #21                            ;
                                     0B  13 002EE         BEQL    63$                                ;
                        18           50  91 002F0         CMPB    R0, #24                            ; 1635
                                     0C  12 002F3         BNEQ    64$                                ;
                        15      02   AA  91 002F5         CMPB    2(ELEM_DESCRIP), #21               ; 1636
                                     06  12 002F9         BNEQ    64$                                ;
                        51      1C   AE  9E 002FB 63$:    MOVAB   NUM_DESCRIP, R1                    ; 1635
                                     04  11 002FF         BRB     65$                                ;
                        51      24   AE  9E 00301 64$:    MOVAB   TEMP_STORE, R1                     ;
                                     51  DD 00305 65$:    PUSHL   R1                                 ;
                        0E           50  91 00307         CMPB    R0, #14                            ; 1616
                                     06  12 0030A         BNEQ    66$                                ;
                        7E      28   AE  3C 0030C         MOVZWL  TEMP_STORE, -(SP)                  ; 1618
                                     1C  11 00310         BRB     70$                                ;
                        18           50  91 00312 66$:    CMPB    R0, #24                            ; 1620
                                     12  12 00315         BNEQ    68$                                ;
                        0E      02   AA  91 00317         CMPB    2(ELEM_DESCRIP), #14               ; 1622
                                     06  12 0031B         BNEQ    67$                                ;
                        51      28   AE  3C 0031D         MOVZWL  TEMP_STORE, R1                     ; 1624
                                     09  11 00321         BRB     69$                                ;
                        51      20   AE  3C 00323 67$:    MOVZWL  NUM_DESCRIP, R1                    ; 1626
                                     03  11 00327         BRB     69$                                ; 1622
                        51           69  3C 00329 68$:    MOVZWL  (R9), R1                           ; 1628
                                     51  DD 0032C 69$:    PUSHL   R1                                 ; 1620
                        18           50  91 0032E 70$:    CMPB    R0, #24                            ; 1608
                                     06  12 00331         BNEQ    71$                                ;
                        7E      02   AA  9A 00333         MOVZBL  2(ELEM_DESCRIP), -(SP)             ;
                                     02  11 00337         BRB     72$                                ;
                                     50  DD 00339 71$:    PUSHL   R0                                 ;
              00000000G 00           04  FB 0033B 72$:    CALLS   #4, BAS$$UDF_WL1                   ; 1606
                                14   AE  D6 00342         INCL    NUM_ELEMS_DONE                     ; 1650
                                     6E  D6 00345         INCL    COLUMN                             ; 1651
                     0C AE           6E  D1 00347         CMPL    COLUMN, UPPER_BOUND1               ; 1653
                                     06  15 0034B         BLEQ    73$                                ;
                                04   AE  D6 0034D         INCL    ROW                                ; 1659
                        6E           01  D0 00350         MOVL    #1, COLUMN                         ; 1660
                                   FDF0  31 00353 73$:    BRW     22$                                ; 1474
                        0E      02   A9  91 00356 74$:    CMPB    2(R9), #14                         ; 1669
                                     0C  13 0035A         BEQL    75$                                ;
                        18      02   A9  91 0035C         CMPB    2(R9), #24                         ; 1670
                                     10  12 00360         BNEQ    76$                                ;
                        0E      02   AA  91 00362         CMPB    2(ELEM_DESCRIP), #14               ; 1671
                                     0A  12 00366         BNEQ    76$                                ;
                                20   AE  9F 00368 75$:    PUSHAB  TEMP_STORE                         ; 1673
              00000000G 00           01  FB 0036B         CALLS   #1, STR$FREE1_DX                   ;
                                         04 00372 76$:    RET                                        ; 1676

; Routine Size:  883 bytes,    Routine Base:  _BAS$CODE + 0026


;  758          1677  1
```

```
1 GLOBAL ROUTINE BAS$OUT_MAT_C (                        ! Matrix print, comma format
1           ARRAY,                                      ! array to print
1           SUBSCRIPT1,                                 ! first optional subscript
1           SUBSCRIPT2                                  ! second optional subscript
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         The array is printed one element at a time with the elements in each row
1 !         being printed in a print zone.  Each row begins on a new line.  Row
1 !         and column zero are not printed.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         ARRAY.rx.a                          ! array to print
1 !         [SUBSCRIPT1.rlu.v]                  ! first optional subscript
1 !         [SUBSCRIPT2.rlu.v]                  ! second optional subscript
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         Signals:
1 !         Data Type Error
1 !
1 !--
1
2     BEGIN
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     BUILTIN
2         ACTUALCOUNT;
2
2     LITERAL
2         V_1D_FLAG = 1,                                ! flag - one dimen. array
2         K_ONE_OPT_ARG = 2,                            ! value of arg. count for one
2                                                       ! optional argument
2         K_TWO_OPT_ARGS = 3,                           ! value of arg. count for two
2                                                       ! optional arguments
2         K_1D = 1;                                     ! one dimension
2
2     LOCAL
2         NUM_ELEMS_DONE,                               ! total number of array elements processed
2         FLAGS,
2              TEMP_STORE : VECTOR [4, LONG],          ! temp storage for calling FETCH_VA
2              ROW,                                    ! current value of subscript 1
2              COLUMN,                                 ! current value of subscript 2
2              UPPER_BOUND1,                           ! upper bound for 1 dimensional
2                                                      ! array and number of rows for 2
2                                                      ! dimensional array
2              TOTAL_NUM_ITEMS,                        ! total number of items in the array
2                                                      ! excluding row and col. 0
2              ELEM_DESCRIP : REF BLOCK [12,BYTE],     ! desc fetched from array
2              NUM_DESCRIP : BLOCK [8,BYTE];           ! numeric desc for FETCH
2
2          MAP
2              ARRAY : REF BLOCK [, BYTE];
2
2          BAS$$CB_GET ();
2      !+
2      ! Check to see if this a list of arrays.  If it is, then print a blank line between
2      ! each array.
2      !-
2
2          IF .CCB [ISB$V_MAT_PRINT] THEN BAS$$BLNK_LINE ();
2
2          CCB [ISB$V_MAT_PRINT] = 1;
2          FLAGS = 0;
2      !+
2      ! Default TEMP_STORE to a dynamic stirng descriptor
2      !-
2          TEMP_STORE [0] = %X'020E0000';
2          TEMP_STORE [1] = %X'00000000';
2      !+
2      ! Check the number of dimensions and set a flag if only one dimension.
2      !-
2
2          IF .ARRAY [DSC$B_DIMCT] EQL K_1D THEN FLAGS = .FLAGS + V_1D_FLAG;
2
2      !+
2      ! Check for optional arguments.  If there are no optional arguments, then set
2      ! the upper bounds based on what is in the descriptor.  If there are optional
2      ! args, then use them as the upper bound.
2      !-
2
2          IF ACTUALCOUNT () LSS K_ONE_OPT_ARG
2          THEN
2
2              IF .ARRAY [DSC$B_DIMCT] EQL K_1D
2              THEN
2      !+
2      ! No optional arguments and a one dimensional array
2      !-
3                  BEGIN
3                  UPPER_BOUND1 = .ARRAY [U1_1D];
3                  TOTAL_NUM_ITEMS = .UPPER_BOUND1;
3                  END
2              ELSE
3                  BEGIN
3      !+
3      ! 2 dimensional array
```

```
3 !-
3               UPPER_BOUND1 = .ARRAY [U2_2D];
3               TOTAL_NUM_ITEMS = .ARRAY [U1_2D]*.UPPER_BOUND1;
2               END;
2
2       IF ACTUALCOUNT () GEQ K_ONE_OPT_ARG
2       THEN
3           BEGIN
3           UPPER_BOUND1 = .SUBSCRIPT1;
3           TOTAL_NUM_ITEMS = .SUBSCRIPT1;
2           END;
2
2       IF ACTUALCOUNT () EQL K_TWO_OPT_ARGS
2       THEN
2 !+
2 ! 2 optional arguments
2 !-
3           BEGIN
3           UPPER_BOUND1 = .SUBSCRIPT2;
3           TOTAL_NUM_ITEMS = .SUBSCRIPT1*.SUBSCRIPT2;
2           END;
2
2 !+
2 ! Initialize the two current subscripts regardless of the number of dimensions
2 !-
2       ROW = COLUMN = NUM_ELEMS_DONE = 1;
2 !+
2 ! Check for array of descriptors.  They could be dynamic string descriptors,
2 ! or numeric descriptors for a dynamically mapped array.  Fetch
2 ! an element (a descriptor) from the array and check the dtype to
2 ! determine if this is a string array or numeric array.
2 !-
2       IF .ARRAY [DSC$E_DTYPE] EQL DSC$K_DTYPE_DSC
2       THEN
3       BEGIN
3
3       NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
3       IF .FLAGS AND V_1D_FLAG
3       THEN
3           ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1)
3       ELSE
3           ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1, 1);
3
3
3       CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
3           SET
3
3           [DSC$K_DTYPE_T] :               ! text
3
3               ;
3
3           [DSC$K_DTYPE_B] :               ! byte
3
4               BEGIN
4               NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
4               NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_B;
4               NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/4;
```

```
;  931          1849  3                 END;
;  932          1850  3
;  933          1851  3             [DSC$K_DTYPE_W] :                  ! word
;  934          1852  3
;  935          1853  4                 BEGIN
;  936          1854  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
;  937          1855  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_W;
;  938          1856  4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/2;
;  939          1857  3                 END;
;  940          1858  3
;  941          1859  3             [DSC$K_DTYPE_L] :                  ! long
;  942          1860  3
;  943          1861  4                 BEGIN
;  944          1862  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
;  945          1863  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_L;
;  946          1864  4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
;  947          1865  3                 END;
;  948          1866  3
;  949          1867  3             [DSC$K_DTYPE_F] :                  ! float
;  950          1868  3
;  951          1869  4                 BEGIN
;  952          1870  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
;  953          1871  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_F;
;  954          1872  4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
;  955          1873  3                 END;
;  956          1874  3
;  957          1875  3             [DSC$K_DTYPE_D] :                  ! double
;  958          1876  3
;  959          1877  4                 BEGIN
;  960          1878  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
;  961          1879  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_D;
;  962          1880  4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
;  963          1881  3                 END;
;  964          1882  3
;  965          1883  3             [DSC$K_DTYPE_G] :                  ! g float
;  966          1884  3
;  967          1885  4                 BEGIN
;  968          1886  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
;  969          1887  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_G;
;  970          1888  4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
;  971          1889  3                 END;
;  972          1890  3
;  973          1891  3             [DSC$K_DTYPE_H] :                  ! h float
;  974          1892  3
;  975          1893  4                 BEGIN
;  976          1894  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
;  977          1895  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_H;
;  978          1896  4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*4;
;  979          1897  3                 END;
;  980          1898  3
;  981          1899  3             [DSC$K_DTYPE_P] :                  ! packed decimal
;  982          1900  3
;  983          1901  4                 BEGIN
;  984          1902  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
;  985          1903  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
;  986          1904  4                 NUM_DESCRIP [DSC$W_LENGTH] = .ARRAY [DSC$W_LENGTH];
;  987          1905  4                 NUM_DESCRIP [DSC$B_SCALE] = .ARRAY [DSC$B_SCALE];
```

```
            END;

        [INRANGE,OUTRANGE] :
            BAS$$STOP (BAS$K_DATTYPEERR);

        TES;

    END;
!+
! Loop thru the array descriptor until all of the elements in the array or as
! specified by the optional arguments have been printed.  Start each row on a
! new line.
!-

    WHILE .NUM_ELEMS_DONE LEQ .TOTAL_NUM_ITEMS DO
        BEGIN
!+
! Based on the data type, JSB or CALL the proper fetch routine to get the element
! out of the array.  The FETCH and STORE routines are called because the array
! may be virtual.
!-

        CASE .ARRAY [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
            SET

            [DSC$K_DTYPE_B] :

                IF .FLAGS AND V_1D_FLAG
                THEN
                    TEMP_STORE [0] = BAS$FET_FA_B_R8 (.ARRAY, .COLUMN)
                ELSE
                    TEMP_STORE [0] = BAS$FET_FA_B_R8 (.ARRAY, .ROW, .COLUMN);

            [DSC$K_DTYPE_W] :

                IF .FLAGS AND V_1D_FLAG
                THEN
                    TEMP_STORE [0] = BAS$FET_FA_W_R8 (.ARRAY, .COLUMN)
                ELSE
                    TEMP_STORE [0] = BAS$FET_FA_W_R8 (.ARRAY, .ROW, .COLUMN);

            [DSC$K_DTYPE_L] :

                IF .FLAGS AND V_1D_FLAG
                THEN
                    TEMP_STORE [0] = BAS$FET_FA_L_R8 (.ARRAY, .COLUMN)
                ELSE
                    TEMP_STORE [0] = BAS$FET_FA_L_R8 (.ARRAY, .ROW, .COLUMN);

            [DSC$K_DTYPE_F] :

                IF .FLAGS AND V_1D_FLAG
                THEN
                    TEMP_STORE [0] = BAS$FET_FA_F_R8 (.ARRAY, .COLUMN)
                ELSE
                    TEMP_STORE [0] = BAS$FET_FA_F_R8 (.ARRAY, .ROW, .COLUMN);

```

```
3       [DSC$K_DTYPE_D] :
3
3           IF .FLAGS AND V_1D_FLAG
3           THEN
3               TEMP_STORE [0] = BAS$FET_FA_D_R8 (.ARRAY, .COLUMN)
3           ELSE
3               TEMP_STORE [0] = BAS$FET_FA_D_R8 (.ARRAY, .ROW, .COLUMN);
3
3       [DSC$K_DTYPE_T] :
3
3           IF .FLAGS AND V_1D_FLAG
3           THEN
3               BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .COLUMN)
3           ELSE
3               BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .ROW, .COLUMN);
3
3       [DSC$K_DTYPE_DSC] :
3
4           BEGIN
4           CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
4               SET
4
4               [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F,
4                DSC$K_DTYPE_D, DSC$K_DTYPE_G, DSC$K_DTYPE_H, DSC$K_DTYPE_P] :
5                   BEGIN
5
5                   TEMP_STORE [0] = %X'00000000';
5                   IF .FLAGS AND V_1D_FLAG
5                   THEN
5                       BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .COLUMN)
5                   ELSE
5                       BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .ROW, .COLUMN);
5
4                   END;
4
4               [DSC$K_DTYPE_T] :
4
4                   IF .FLAGS AND V_1D_FLAG
4                   THEN
4                       BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .COLUMN)
4                   ELSE
4                       BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .ROW, .COLUMN);
4
4               [INRANGE,OUTRANGE] :
4                   BAS$$STOP (BAS$K_DATTYPERR);
4
4               TES;
4
3           END;                                ! end of dtype dsc
3
3       [DSC$K_DTYPE_P] :
3       !+
3       ! Must pass a descriptor to BAS$$UDF_WL1.  Construct a class SD
3       ! descriptor here, and set the pointer field to TEMP_STORE.
3       !-
4           BEGIN
4           NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
```

```
; 1102          2020  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
; 1103          2021  4                 NUM_DESCRIP [DSC$W_LENGTH] = .ARRAY [DSC$W_LENGTH];
; 1104          2022  4                 NUM_DESCRIP [DSC$B_SCALE] = .ARRAY [DSC$B_SCALE];
; 1105          2023  4                 NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
; 1106          2024  4
; 1107          2025  4                 IF .FLAGS AND V_1D_FLAG
; 1108          2026  4                 THEN
; 1109          2027  4                     BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .COLUMN)
; 1110          2028  4                 ELSE
; 1111          2029  4                     BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .ROW, .COLUMN);
; 1112          2030  3                 END;
; 1113          2031  3
; 1114          2032  3             [DSC$K_DTYPE_G] :
; 1115          2033  3
; 1116          2034  3                 IF .FLAGS AND V_1D_FLAG
; 1117          2035  3                 THEN
; 1118          2036  3                     TEMP_STORE [0] = BAS$FET_FA_G_R8 (.ARRAY, .COLUMN)
; 1119          2037  3                 ELSE
; 1120          2038  3                     TEMP_STORE [0] = BAS$FET_FA_G_R8 (.ARRAY, .ROW, .COLUMN);
; 1121          2039  3
; 1122          2040  3             [DSC$K_DTYPE_H] :
; 1123          2041  3
; 1124          2042  3                 IF .FLAGS AND V_1D_FLAG
; 1125          2043  3                 THEN
; 1126          2044  3                     TEMP_STORE [0] = BAS$FET_FA_H_R8 (.ARRAY, .COLUMN)
; 1127          2045  3                 ELSE
; 1128          2046  3                     TEMP_STORE [0] = BAS$FET_FA_H_R8 (.ARRAY, .ROW, .COLUMN);
; 1129          2047  3
; 1130          2048  3             [INRANGE, OUTRANGE] :
; 1131          2049  3                 BAS$$STOP (BAS$K_DATTYPERR);
; 1132          2050  3             TES;
; 1133          2051  3
; 1134          2052  3         BAS$$UDF_WL1 (
; 1135          2053  4             BEGIN
; 1136          2054  4
; 1137          2055  4             IF (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC) THEN .ELEM_DESCRIP [DSC$B_DTYPE] ELSE .ARRAY [DSC$
; 1138          2056  4
; 1139          2057  4             END
; 1140          2058  3         ,                                       !
; 1141          2059  4             BEGIN
; 1142          2060  4             MAP
; 1143          2061  4                 TEMP_STORE : BLOCK [8,BYTE];
; 1144          2062  4
; 1145          2063  5             (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T
; 1146          2064  5              THEN
; 1147          2065  5                 .TEMP_STORE [DSC$W_LENGTH]
; 1148          2066  5              ELSE
; 1149          2067  6                 (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC
; 1150          2068  6                  THEN
; 1151          2069  6                     IF .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T
; 1152          2070  6                     THEN
; 1153          2071  6                         .TEMP_STORE [DSC$W_LENGTH]
; 1154          2072  6                     ELSE
; 1155          2073  6                         .NUM_DESCRIP [DSC$W_LENGTH]
; 1156          2074  6                  ELSE
; 1157          2075  5                     .ARRAY [DSC$W_LENGTH]))
; 1158          2076  4             END
; 1159          2077 3          ,                                        !
; 1160          2078 4              (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
; 1161          2079 4               THEN
; 1162          2080 4                  NUM_DESCRIP                      ! pass dsc for packed
; 1163          2081 4               ELSE
; 1164          2082 5                  (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
; 1165          2083 5                      .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
; 1166          2084 5                   THEN
; 1167          2085 5                      NUM_DESCRIP
; 1168          2086 5                   ELSE
; 1169          2087 3                      TEMP_STORE)),
; 1170          2088 3
; 1171          2089 3          ! If this is the last element of this row, then pass the "no format"
; 1172          2090 3          ! argument so that the first element of the next row starts on a
; 1173          2091 3          ! new line
; 1174          2092 3              (IF .COLUMN EQL .UPPER_BOUND1 THEN BAS$K_NO_FORM ELSE BAS$K_COMMA_FOR));
; 1175          2093 3          NUM_ELEMS_DONE = .NUM_ELEMS_DONE + 1;
; 1176          2094 3          COLUMN = .COLUMN + 1;
; 1177          2095 3
; 1178          2096 3          IF .COLUMN GTR .UPPER_BOUND1
; 1179          2097 3          THEN
; 1180          2098 4              BEGIN
; 1181          2099 4 !+
; 1182          2100 4 ! It is time to start a new row.
; 1183          2101 4 !-
; 1184          2102 4              ROW = .ROW + 1;
; 1185          2103 4              COLUMN = 1;
; 1186          2104 3              END;
; 1187          2105 3
; 1188          2106 2          END;                                     ! end of the WHILE loop
; 1189          2107 2
; 1190          2108 2 !+
; 1191          2109 2 ! Return any temporary storage used and then return
; 1192          2110 2 !-
; 1193          2111 2
; 1194          2112 2      IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T OR
; 1195          2113 3        (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
; 1196          2114 3          .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T)
; 1197          2115 2      THEN
; 1198          2116 2          STR$FREE1_DX (TEMP_STORE);
; 1199          2117 2
; 1200          2118 2      RETURN;
; 1201          2119 1      END;                                         !End of BAS$OUT_MAT_C


                                  0FFC 00000         .ENTRY  BAS$OUT_MAT_C, Save R2,R3,R4,R5,R6,R7,R8,-  ; 1678
                                                             R9,R10,R11                                   ;
                5E            30   C2 00002          SUBL2   #48, SP                                      ;
                   00000000G  00   16 00005          JSB     BAS$$CB_GET                                  ; 1749
   07      97   AB            02   E1 0000B          BBC     #2, -105(CCB), 1$                            ; 1755
     00000000G  00            00   FB 00010          CALLS   #0, BAS$$BLNK_LINE                           ;
           97   AB            04   88 00017 1$:      BISB2   #4, -105(CCB)                                ; 1757
                        08    AE   D4 0001B          CLRL    FLAGS                                        ; 1758
           20   AE 020E0000   8F   D0 0001E          MOVL    #34471936, TEMP_STORE                        ; 1762
```

```
                                        24   AE  D4 00026        CLRL    TEMP_STORE+4                                 ; 1763
                              59        04   AC  D0 00029        MOVL    ARRAY, R9                                    ; 1768
                                             50  D4 0002D        CLRL    R0                                           ;
                              01        0B   A9  91 0002F        CMPB    11(R9), #1                                   ;
                                             05  12 00033        BNEQ    2$                                           ;
                                             50  D6 00035        INCL    R0                                           ;
                                        08   AE  D6 00037        INCL    FLAGS                                        ;
                              02             6C  91 0003A 2$:    CMPB    (AP), #2                                     ; 1776
                                             1B  1E 0003D        BGEQU   4$                                           ;
                              0C             50  E9 0003F        BLBC    R0, 3$                                       ; 1779
                         0C   AE        1C   A9  D0 00042        MOVL    28(R9), UPPER_BOUND1                         ; 1785
                         10   AE        0C   AE  D0 00047        MOVL    UPPER_BOUND1, TOTAL_NUM_ITEMS                ; 1786
                                             0C  11 0004C        BRB     4$                                           ; 1779
                         0C   AE        28   A9  D0 0004E 3$:    MOVL    40(R9), UPPER_BOUND1                         ; 1793
          10   AE        20   A9        0C   AE  C5 00053        MULL3   UPPER_BOUND1, 32(R9), TOTAL_NUM_ITEMS        ; 1794
                              02             6C  91 0005A 4$:    CMPB    (AP), #2                                     ; 1797
                                             0A  1F 0005D        BLSSU   5$                                           ;
                         0C   AE        08   AC  D0 0005F        MOVL    SUBSCRIPT1, UPPER_BOUND1                     ; 1800
                         10   AE        08   AC  D0 00064        MOVL    SUBSCRIPT1, TOTAL_NUM_ITEMS                  ; 1801
                              03             6C  91 00069 5$:    CMPB    (AP), #3                                     ; 1804
                                             0C  12 0006C        BNEQ    6$                                           ;
                         0C   AE        0C   AC  D0 0006E        MOVL    SUBSCRIPT2, UPPER_BOUND1                     ; 1810
          10   AE        08   AC        0C   AC  C5 00073        MULL3   SUBSCRIPT2, SUBSCRIPT1, TOTAL_NUM_ITEMS      ; 1811
                         14   AE             01  D0 0007A 6$:    MOVL    #1, NUM_ELEMS_DONE                           ; 1817
                              6E             01  D0 0007E        MOVL    #1, COLUMN                                   ;
                         04   AE             01  D0 00081        MOVL    #1, ROW                                      ;
                              18        02   A9  91 00085        CMPB    2(R9), #24                                   ; 1824
                                             78  12 00089        BNEQ    13$                                          ;
                         1C   AE        20   AE  9E 0008B        MOVAB   TEMP_STORE, NUM_DESCRIP+4                    ; 1828
                              0D        08   AE  E9 00090        BLBC    FLAGS, 7$                                    ; 1829
                                             01  DD 00094        PUSHL   #1                                           ; 1831
                                             59  DD 00096        PUSHL   R9                                           ;
                   00000000G  00             02  FB 00098        CALLS   #2, BAS$FETCH_DESC                           ;
                                             0D  11 0009F        BRB     8$                                           ;
                                             01  DD 000A1 7$:    PUSHL   #1                                           ; 1833
                                             01  DD 000A3        PUSHL   #1                                           ;
                                             59  DD 000A5        PUSHL   R9                                           ;
                   00000000G  00             03  FB 000A7        CALLS   #3, BAS$FETCH_DESC                           ;
                              5A             50  D0 000AE 8$:    MOVL    R0, ELEM_DESCRIP                             ;
                   16         06        02   AA  8F 000B1        CASEB   2(ELEM_DESCRIP), #6, #22                     ; 1836
   002E            004F       0045         003B     000B6 9$:    .WORD   11$-9$,-                                     ;
   002E            002E       0063         0057     000BE                12$-9$,-                                     ;
   002E            002E       002E         0090     000C6                14$-9$,-                                     ;
   0081            002E       002E         002E     000CE                10$-9$,-                                     ;
   002E            002E       002E         002E     000D6                15$-9$,-                                     ;
                   0077       006B         002E     000DE                17$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
                                                                         22$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
                                                                         21$-9$,-                                     ;
                                                                         10$-9$,-                                     ;
```

```
                                                                          10$-9$,-                          ;
                                                                          10$-9$,-                          ;
                                                                          10$-9$,-                          ;
                                                                          10$-9$,-                          ;
                                                                          18$-9$,-                          ;
                                                                          20$-9$                            ;
                          7E        00G  8F   9A 000E4 10$:      MOVZBL   #BAS$K_DATTYPERR, -(SP)           ; 1909
               00000000G  00             01   FB 000E8           CALLS    #1, BAS$$STOP                     ;
                                         55   11 000EF           BRB      22$                               ;
                     18   AE 01060001    8F   D0 000F1 11$:      MOVL     #17170433, NUM_DESCRIP            ; 1848
                                         4B   11 000F9           BRB      22$                               ; 1836
                     18   AE 01070002    8F   D0 000FB 12$:      MOVL     #17235970, NUM_DESCRIP            ; 1856
                                         41   11 00103 13$:      BRB      22$                               ; 1836
                     1A   AE       0108  8F   B0 00105 14$:      MOVW     #264, NUM_DESCRIP+2               ; 1863
                                         06   11 0010B           BRB      16$                               ; 1864
                     1A   AE       010A  8F   B0 0010D 15$:      MOVW     #266, NUM_DESCRIP+2               ; 1871
                     18   AE             04   B0 00113 16$:      MOVW     #4, NUM_DESCRIP                   ; 1872
                                         2D   11 00117           BRB      22$                               ; 1836
                     1A   AE       010B  8F   B0 00119 17$:      MOVW     #267, NUM_DESCRIP+2               ; 1879
                                         06   11 0011F           BRB      19$                               ; 1880
                     1A   AE       011B  8F   B0 00121 18$:      MOVW     #283, NUM_DESCRIP+2               ; 1887
                     18   AE             08   B0 00127 19$:      MOVW     #8, NUM_DESCRIP                   ; 1888
                                         19   11 0012B           BRB      22$                               ; 1836
                     18   AE 011C0010    8F   D0 0012D 20$:      MOVL     #18612240, NUM_DESCRIP            ; 1896
                                         0F   11 00135           BRB      22$                               ; 1836
                     1A   AE       0915  8F   B0 00137 21$:      MOVW     #2325, NUM_DESCRIP+2              ; 1903
                     18   AE             69   B0 0013D           MOVW     (R9), NUM_DESCRIP                 ; 1904
                     20   AE         08  A9   90 00141           MOVB     8(R9), NUM_DESCRIP+8              ; 1905
                     10   AE         14  AE   D1 00146 22$:      CMPL     NUM_ELEMS_DONE, TOTAL_NUM_ITEMS   ; 1920
                                         03   15 0014B           BLEQ     23$                               ;
                                       0206   31 0014D           BRW      74$                               ;
              16          06         02  A9   8F 00150 23$:      CASEB    2(R9), #6, #22                    ; 1928
   00F7       0067      004C           0031      00155 24$:      .WORD    25$-24$,-                         ;
   00F7       00F7      009D           0082      0015D                    28$-24$,-                         ;
   00F7       00F7      00F7           00B9      00165                    31$-24$,-                         ;
   0113       00F7      00F7           00F7      0016D                    44$-24$,-                         ;
   00F7       00C4      00F7           00F7      00175                    34$-24$,-                         ;
              0169      014E           00F7      0017D                    37$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          41$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          47$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          42$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          44$-24$,-                         ;
                                                                          53$-24$,-                         ;
                                                                          56$-24$                           ;
                                       00C6   31 00183           BRW      44$                               ; 2049
                          05         08  AE   E9 00186 25$:      BLBC     FLAGS, 26$                        ; 1933
```

```
                              51            6E  D0 0018A         MOVL    COLUMN, R1                        ; 1935
                                            07  11 0018D         BRB     27$                               ;
                              52            6E  D0 0018F 26$:    MOVL    COLUMN, R2                        ; 1937
                              51      04    AE  D0 00192         MOVL    ROW, R1                           ;
                              50            59  D0 00196 27$:    MOVL    R9, R0                            ;
                                 00000000G  00  16 00199         JSB     BAS$FET_FA_B_R8                   ;
                                            6A  11 0019F         BRB     40$                               ;
                              05      08    AE  E9 001A1 28$:    BLBC    FLAGS, 29$                        ; 1941
                              51            6E  D0 001A5         MOVL    COLUMN, R1                        ; 1943
                                            07  11 001A8         BRB     30$                               ;
                              52            6E  D0 001AA 29$:    MOVL    COLUMN, R2                        ; 1945
                              51      04    AE  D0 001AD         MOVL    ROW, R1                           ;
                              50            59  D0 001B1 30$:    MOVL    R9, R0                            ;
                                 00000000G  00  16 001B4         JSB     BAS$FET_FA_W_R8                   ;
                                            4F  11 001BA         BRB     40$                               ;
                              05      08    AE  E9 001BC 31$:    BLBC    FLAGS, 32$                        ; 1949
                              51            6E  D0 001C0         MOVL    COLUMN, R1                        ; 1951
                                            07  11 001C3         BRB     33$                               ;
                              52            6E  D0 001C5 32$:    MOVL    COLUMN, R2                        ; 1953
                              51      04    AE  D0 001C8         MOVL    ROW, R1                           ;
                              50            59  D0 001CC 33$:    MOVL    R9, R0                            ;
                                 00000000G  00  16 001CF         JSB     BAS$FET_FA_L_R8                   ;
                                            34  11 001D5         BRB     40$                               ;
                              05      08    AE  E9 001D7 34$:    BLBC    FLAGS, 35$                        ; 1957
                              51            6E  D0 001DB         MOVL    COLUMN, R1                        ; 1959
                                            07  11 001DE         BRB     36$                               ;
                              52            6E  D0 001E0 35$:    MOVL    COLUMN, R2                        ; 1961
                              51      04    AE  D0 001E3         MOVL    ROW, R1                           ;
                              50            59  D0 001E7 36$:    MOVL    R9, R0                            ;
                                 00000000G  00  16 001EA         JSB     BAS$FET_FA_F_R8                   ;
                                            19  11 001F0         BRB     40$                               ;
                              05      08    AE  E9 001F2 37$:    BLBC    FLAGS, 38$                        ; 1965
                              51            6E  D0 001F6         MOVL    COLUMN, R1                        ; 1967
                                            07  11 001F9         BRB     39$                               ;
                              52            6E  D0 001FB 38$:    MOVL    COLUMN, R2                        ; 1969
                              51      04    AE  D0 001FE         MOVL    ROW, R1                           ;
                              50            59  D0 00202 39$:    MOVL    R9, R0                            ;
                                 00000000G  00  16 00205         JSB     BAS$FET_FA_D_R8                   ;
                                          00C9  31 0020B 40$:    BRW     59$                               ;
                              4C      08    AE  E9 0020E 41$:    BLBC    FLAGS, 46$                        , 1973
                                            6E  DD 00212         PUSHL   COLUMN                            ; 1975
                                      24    AE  9F 00214         PUSHAB  TEMP_STORE                        ;
                                            6C  11 00217         BRB     49$                               ;
                 16           06      02    AA  8F 00219 42$:    CASEB   2(ELEM_DESCRIP), #6, #22          ; 1982
       002E      003B         003B        003B     0021E 43$:    .WORD   45$-43$,-                         ;
       002E      002E         003B        003B     00226                 45$-43$,-                         ;
       002E      002E         002E        FFF0     0022E                 45$-43$,-                         ;
       003B      002E         002E        002E     00236                 44$-43$,-                         ;
       002E      002E         002E        002E     0023E                 45$-43$,-                         ;
                 003B         003B        002E     00246                 45$-43$,-                         ;
                                                                         44$-43$,-                         ;
                                                                         44$-43$,-                         ;
                                                                         41$-43$,-                         ;
                                                                         44$-43$,-                         ;
                                                                         44$-43$,-                         ;
                                                                         44$-43$,-                         ;
                                                                         44$-43$,-                         ;
```

```
                                                              44$-43$,-                          ;
                                                              44$-43$,-                          ;
                                                              45$-43$,-                          ;
                                                              44$-43$,-                          ;
                                                              44$-43$,-                          ;
                                                              44$-43$,-                          ;
                                                              44$-43$,-                          ;
                                                              44$-43$,-                          ;
                                                              45$-43$,-                          ;
                                                              45$-43$                            ;
            7E        00G  8F  9A 0024C 44$:     MOVZBL  #BAS$K_DATTYPERR, -(SP)                 ; 2007
 00000000G  00             01  FB 00250          CALLS   #1, BAS$$STOP                           ;
                           48  11 00257          BRB     52$                                     ;
                      20   AE  D4 00259 45$:     CLRL    TEMP_STORE                              ; 1989
                           1E  11 0025C          BRB     48$                                     ; 1990
                           6E  DD 0025E 46$:     PUSHL   COLUMN                                  ; 2004
                      08   AE  DD 00260          PUSHL   ROW                                     ;
                      28   AE  9F 00263          PUSHAB  TEMP_STORE                              ;
                           30  11 00266          BRB     51$                                     ;
       1A   AE      0915   8F  B0 00268 47$:     MOVW    #2325, NUM_DESCRIP+2                    ; 2020
       18   AE             69  B0 0026E          MOVW    (R9), NUM_DESCRIP                       ; 2021
       20   AE        08   A9  90 00272          MOVB    8(R9), NUM_DESCRIP+8                    ; 2022
       1C   AE        20   AE  9E 00277          MOVAB   TEMP_STORE, NUM_DESCRIP+4               ; 2023
            10        08   AE  E9 0027C 48$:     BLBC    FLAGS, 50$                              ; 2025
                           6E  DD 00280          PUSHL   COLUMN                                  ; 2027
                      1C   AE  9F 00282          PUSHAB  NUM_DESCRIP                             ;
                           59  DD 00285 49$:     PUSHL   R9                                      ;
 00000000G  00             03  FB 00287          CALLS   #3, BAS$FETCH_BFA                       ;
                           48  11 0028E          BRB     60$                                     ;
                           6E  DD 00290 50$:     PUSHL   COLUMN                                  ; 2029
                      08   AE  DD 00292          PUSHL   ROW                                     ;
                      20   AE  9F 00295          PUSHAB  NUM_DESCRIP                             ;
                           59  DD 00298 51$:     PUSHL   R9                                      ;
 00000000G  00             04  FB 0029A          CALLS   #4, BAS$FETCH_BFA                       ;
                           38  11 002A1 52$:     BRB     60$                                     ; 1928
            05        08   AE  E9 002A3 53$:     BLBC    FLAGS, 54$                              ; 2034
            51             6E  D0 002A7          MOVL    COLUMN, R1                              ; 2036
                           07  11 002AA          BRB     55$                                     ;
            52             6E  D0 002AC 54$:     MOVL    COLUMN, R2                              ; 2038
            51        04   AE  D0 002AF          MOVL    ROW, R1                                 ;
            50             59  D0 002B3 55$:     MOVL    R9, R0                                  ;
                 00000000G 00  16 002B6          JSB     BAS$FET_FA_G_R8                         ;
                           19  11 002BC          BRB     59$                                     ;
            05        08   AE  E9 002BE 56$:     BLBC    FLAGS, 57$                              ; 2042
            51             6E  D0 002C2          MOVL    COLUMN, R1                              ; 2044
                           07  11 002C5          BRB     58$                                     ;
            52             6E  D0 002C7 57$:     MOVL    COLUMN, R2                              ; 2046
            51        04   AE  D0 002CA          MOVL    ROW, R1                                 ;
            50             59  D0 002CE 58$:     MOVL    R9, R0                                  ;
                 00000000G 00  16 002D1          JSB     BAS$FET_FA_H_R8                         ;
       20   AE             50  D0 002D7 59$:     MOVL    R0, TEMP_STORE                          ;
       0C   AE             6E  D1 002DB 60$:     CMPL    COLUMN, UPPER_BOUND1                    ; 2092
                           04  12 002DF          BNEQ    61$                                     ;
                           03  DD 002E1          PUSHL   #3                                      ;
                           02  11 002E3          BRB     62$                                     ;
                           02  DD 002E5 61$:     PUSHL   #2                                      ;
            50        02   A9  9A 002E7 62$:     MOVZBL  2(R9), R0                               ; 2078
```

```
                    15          50  91 002EB        CMPB    R0, #21                          ;
                                0B  13 002EE        BEQL    63$                              ;
                    18          50  91 002F0        CMPB    R0, #24                          ; 2082
                                0C  12 002F3        BNEQ    64$                              ;
                    15     02   AA  91 002F5        CMPB    2(ELEM_DESCRIP), #21             ; 2083
                                06  12 002F9        BNEQ    64$                              ;
                    51     1C   AE  9E 002FB 63$:   MOVAB   NUM_DESCRIP, R1                  ; 2082
                                04  11 002FF        BRB     65$                              ;
                    51     24   AE  9E 00301 64$:   MOVAB   TEMP_STORE, R1                   ;
                                51  DD 00305 65$:   PUSHL   R1                               ;
                    0E          50  91 00307        CMPB    R0, #14                          ; 2063
                                06  12 0030A        BNEQ    66$                              ;
                    7E     28   AE  3C 0030C        MOVZWL  TEMP_STORE, -(SP)                ; 2065
                                1C  11 00310        BRB     70$                              ;
                    18          50  91 00312 66$:   CMPB    R0, #24                          ; 2067
                                12  12 00315        BNEQ    68$                              ;
                    0E     02   AA  91 00317        CMPB    2(ELEM_DESCRIP), #14             ; 2069
                                06  12 0031B        BNEQ    67$                              ;
                    51     28   AE  3C 0031D        MOVZWL  TEMP_STORE, R1                   ; 2071
                                09  11 00321        BRB     69$                              ;
                    51     20   AE  3C 00323 67$:   MOVZWL  NUM_DESCRIP, R1                  ; 2073
                                03  11 00327        BRB     69$                              ; 2069
                    51          69  3C 00329 68$:   MOVZWL  (R9), R1                         ; 2075
                                51  DD 0032C 69$:   PUSHL   R1                               ; 2067
                    18          50  91 0032E 70$:   CMPB    R0, #24                          ; 2055
                                06  12 00331        BNEQ    71$                              ;
                    7E     02   AA  9A 00333        MOVZBL  2(ELEM_DESCRIP), -(SP)           ;
                                02  11 00337        BRB     72$                              ;
                                50  DD 00339 71$:   PUSHL   R0                               ;
           00000000G  00        04  FB 0033B 72$:   CALLS   #4, BAS$$UDF_WL1                 ; 2053
                           14   AE  D6 00342        INCL    NUM_ELEMS_DONE                   ; 2093
                                6E  D6 00345        INCL    COLUMN                           ; 2094
                  0C  AE        6E  D1 00347        CMPL    COLUMN, UPPER_BOUND1             ; 2096
                                06  15 0034B        BLEQ    73$                              ;
                           04   AE  D6 0034D        INCL    ROW                              ; 2102
                    6E          01  D0 00350        MOVL    #1, COLUMN                       ; 2103
                              FDF0  31 00353 73$:   BRW     22$                              ; 1920
                    0E     02   A9  91 00356 74$:   CMPB    2(R9), #14                       ; 2112
                                0C  13 0035A        BEQL    75$                              ;
                    18     02   A9  91 0035C        CMPB    2(R9), #24                       ; 2113
                                10  12 00360        BNEQ    76$                              ;
                    0E     02   AA  91 00362        CMPB    2(ELEM_DESCRIP), #14             ; 2114
                                0A  12 00366        BNEQ    76$                              ;
                           20   AE  9F 00368 75$:   PUSHAB  TEMP_STORE                       ; 2116
           00000000G  00        01  FB 0036B        CALLS   #1, STR$FREE1_DX                 ;
                                04  00372 76$:      RET                                      ; 2119

; Routine Size:  883 bytes,    Routine Base:  _BAS$CODE + 0399

; 1202          2120  1
```

```
1 GLOBAL ROUTINE BAS$OUT_MAT_B (                         ! Matrix print, no format
1           ARRAY,                                       ! array to print
1           SUBSCRIPT1,                                  ! first optional subscript
1           SUBSCRIPT2                                   ! second optional subscript
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         The array is printed one element at a time with each element
1 !         being printed on a separate line.  Row and column zero are not printed.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         ARRAY.rx.a                        ! array to print
1 !         [SUBSCRIPT1.rlu.v]                ! first optional subscript
1 !         [SUBSCRIPT2.rlu.v]                ! second optional subscript
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         Signals:
1 !         Data Type Error
1 !
1 !--
1
2     BEGIN
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     BUILTIN
2         ACTUALCOUNT;
2
2     LITERAL
2         V_1D_FLAG = 1,                        ! flag - one dimen. array
2         K_ONE_OPT_ARG = 2,                    ! value of arg. count for one
2                                               ! optional argument
2         K_TWO_OPT_ARGS = 3,                   ! value of arg. count for two
2                                               ! optional arguments
2         K_1D = 1;                             ! one dimension
2
2     LOCAL
2         NUM_ELEMS_DONE,                       ! total number of array elements processed
2         FLAGS,
2         TEMP_STORE : VECTOR [4, LONG],        ! temp storage for calling FETCH_VA
```

```
2          ROW,                                    ! current value of subscript 1
2          COLUMN,                                 ! current value of subscript 2
2          UPPER_BOUND1,                           ! upper bound for 1 dimensional
2                                                  ! array and number of rows for 2
2                                                  ! dime-sional array
2          TOTAL_NUM_ITEMS,                        . total number of items in the array
2                                                  ! excluding row and col. 0
2          ELEM_DESCRIP : REF BLOCK [12,BYTE],     ! desc fetched from array
2          NUM_DESCRIP : BLOCK [8,BYTE];           ! numeric desc for FETCH
2
2      MAP
2          ARRAY : REF BLOCK [, BYTE];
2
2      BAS$$CB_GET ();
2  !+
2  ! Check to see if this a list of arrays.  If it is, then print a blank line between
2  ! each array.
2  !-
2
2      IF .CCB [ISB$V_MAT_PRINT] THEN BAS$$BLNK_LINE ();
2
2      CCB [ISB$V_MAT_PRINT] = 1;
2      FLAGS = 0;
2  !+
2  ! Default TEMP_STORE to a dynamic stirng descriptor
2  !-
2      TEMP_STORE [0] = %X'020E0000';
2      TEMP_STORE [1] = %X'00000000';
2  !+
2  ! Check the number of dimensions and set a flag if only one dimension.
2  !-
2
2      IF .ARRAY [DSC$B_DIMCT] EQL K_1D THEN FLAGS = .FLAGS + V_1D_FLAG;
2
2  !+
2  ! Check for optional arguments.  If there are no optional arguments, then set
2  ! the upper bounds based on what is in the descriptor.  If there are optional
2  ! args, then use them as the upper bound.
2  !-
2
2      IF ACTUALCOUNT () LSS K_ONE_OPT_ARG
2      THEN
2
2          IF .ARRAY [DSC$B_DIMCT] EQL K_1D
2          THEN
2  !+
2  ! No optional arguments and a one dimensional array
2  !-
3              BEGIN
3              UPPER_BOUND1 = .ARRAY [U1_1D];
3              TOTAL_NUM_ITEMS = .UPPER_BOUND1;
3              END
2          ELSE
3              BEGIN
3  !+
3  ! 2 dimensional array
3  !-
```

```
; 1318          2235  3                 UPPER_BOUND1 = .ARRAY [U2_2D];
; 1319          2236  3                 TOTAL_NUM_ITEMS = .ARRAY [U1_2D]*.UPPER_BOUND1;
; 1320          2237  2                 END;
; 1321          2238  2
; 1322          2239  2         IF ACTUALCOUNT () GEQ K_ONE_OPT_ARG
; 1323          2240  2         THEN
; 1324          2241  3             BEGIN
; 1325          2242  3             UPPER_BOUND1 = .SUBSCRIPT1;
; 1326          2243  3             TOTAL_NUM_ITEMS = .SUBSCRIPT1;
; 1327          2244  2             END;
; 1328          2245  2
; 1329          2246  2         IF ACTUALCOUNT () EQL K_TWO_OPT_ARGS
; 1330          2247  2         THEN
; 1331          2248  2   !+
; 1332          2249  2   ! 2 optional arguments
; 1333          2250  2   !-
; 1334          2251  3             BEGIN
; 1335          2252  3             UPPER_BOUND1 = .SUBSCRIPT2;
; 1336          2253  3             TOTAL_NUM_ITEMS = .SUBSCRIPT1*.SUBSCRIPT2;
; 1337          2254  2             END;
; 1338          2255  2
; 1339          2256  2   !+
; 1340          2257  2   ! Initialize the two current subscripts regardless of the number of dimensions
; 1341          2258  2   !-
; 1342          2259  2         ROW = COLUMN = NUM_ELEMS_DONE = 1;
; 1343          2260  2   !+
; 1344          2261  2   ! Check for array of descriptors.  They could be dynamic string descriptors,
; 1345          2262  2   ! or numeric descriptors for a dynamically mapped array.  Fetch
; 1346          2263  2   ! an element (a descriptor) from the array and check the dtype to
; 1347          2264  2   ! determine if this is a string array or numeric array.
; 1348          2265  2   !-
; 1349          2266  2
; 1350          2267  2         IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC
; 1351          2268  2         THEN
; 1352          2269  3         BEGIN
; 1353          2270  3
; 1354          2271  3         NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
; 1355          2272  3         IF .FLAGS AND V_1D_FLAG
; 1356          2273  3         THEN
; 1357          2274  3             ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1)
; 1358          2275  3         ELSE
; 1359          2276  3             ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1, 1);
; 1360          2277  3
; 1361          2278  3
; 1362          2279  3         CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
; 1363          2280  3             SET
; 1364          2281  3
; 1365          2282  3             [DSC$K_DTYPE_T] :                 ! text
; 1366          2283  3
; 1367          2284  3                 ;
; 1368          2285  3
; 1369          2286  3             [DSC$K_DTYPE_B] :                 ! byte
; 1370          2287  3
; 1371          2288  4                 BEGIN
; 1372          2289  4                 NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
; 1373          2290  4                 NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_B;
; 1374          2291  4                 NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/4;
; 1375          2292  3             END;
; 1376          2293  3
; 1377          2294  3         [DSC$K_DTYPE_W] :                   ! word
; 1378          2295  3
; 1379          2296  4             BEGIN
; 1380          2297  4             NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
; 1381          2298  4             NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_W;
; 1382          2299  4             NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/2;
; 1383          2300  3             END;
; 1384          2301  3
; 1385          2302  3         [DSC$K_DTYPE_L] :                   ! long
; 1386          2303  3
; 1387          2304  4             BEGIN
; 1388          2305  4             NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
; 1389          2306  4             NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_L;
; 1390          2307  4             NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
; 1391          2308  3             END;
; 1392          2309  3
; 1393          2310  3         [DSC$K_DTYPE_F] :                   ! float
; 1394          2311  3
; 1395          2312  4             BEGIN
; 1396          2313  4             NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
; 1397          2314  4             NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_F;
; 1398          2315  4             NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
; 1399          2316  3             END;
; 1400          2317  3
; 1401          2318  3         [DSC$K_DTYPE_D] :                   ! double
; 1402          2319  3
; 1403          2320  4             BEGIN
; 1404          2321  4             NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
; 1405          2322  4             NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_D;
; 1406          2323  4             NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
; 1407          2324  3             END;
; 1408          2325  3
; 1409          2326  3         [DSC$K_DTYPE_G] :                   ! g float
; 1410          2327  3
; 1411          2328  4             BEGIN
; 1412          2329  4             NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
; 1413          2330  4             NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_G;
; 1414          2331  4             NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
; 1415          2332  3             END;
; 1416          2333  3
; 1417          2334  3         [DSC$K_DTYPE_H] :                   ! h float
; 1418          2335  3
; 1419          2336  4             BEGIN
; 1420          2337  4             NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
; 1421          2338  4             NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_H;
; 1422          2339  4             NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*4;
; 1423          2340  3             END;
; 1424          2341  3
; 1425          2342  3         [DSC$K_DTYPE_P] :                   ! packed decimal
; 1426          2343  3
; 1427          2344  4             BEGIN
; 1428          2345  4             NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
; 1429          2346  4             NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
; 1430          2347  4             NUM_DESCRIP [DSC$W_LENGTH] = .ARRAY [DSC$W_LENGTH];
; 1431          2348  4             NUM_DESCRIP [DSC$B_SCALE] = .ARRAY [DSC$B_SCALE];
```

```
3                   END;
3
3               [INRANGE,OUTRANGE] :
3                   BAS$$STOP (BAS$K_DATTYPERR);
3
3               TES;
3
2           END;
2   !+
2   ! Loop thru the array descriptor until all of the elements in the array or as
2   ! specified by the optional arguments have been printed.  Start each row on a
2   ! new line.
2   !-
2
2       WHILE .NUM_ELEMS_DONE LEQ .TOTAL_NUM_ITEMS DO
3           BEGIN
3   !+
3   ! Based on the data type, JSB or CALL the proper fetch routine to get the element
3   ! out of the array.  The FETCH and STORE routines are called because the array
3   ! may be virtual.
3   !-
3
3           CASE .ARRAY [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
3               SET
3
3               [DSC$K_DTYPE_B] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       TEMP_STORE [0] = BAS$FET_FA_B_R8 (.ARRAY, .COLUMN)
3                   ELSE
3                       TEMP_STORE [0] = BAS$FET_FA_B_R8 (.ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_W] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       TEMP_STORE [0] = BAS$FET_FA_W_R8 (.ARRAY, .COLUMN)
3                   ELSE
3                       TEMP_STORE [0] = BAS$FET_FA_W_R8 (.ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_L] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       TEMP_STORE [0] = BAS$FET_FA_L_R8 (.ARRAY, .COLUMN)
3                   ELSE
3                       TEMP_STORE [0] = BAS$FET_FA_L_R8 (.ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_F] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       TEMP_STORE [0] = BAS$FET_FA_F_R8 (.ARRAY, .COLUMN)
3                   ELSE
3                       TEMP_STORE [0] = BAS$FET_FA_F_R8 (.ARRAY, .ROW, .COLUMN);
3
```

```
3        [DSC$K_DTYPE_D] :
3
3            IF .FLAGS AND V_1D_FLAG
3            THEN
3                TEMP_STORE [0] = BAS$FET_FA_D_R8 (.ARRAY, .COLUMN)
3            ELSE
3                TEMP_STORE [0] = BAS$FET_FA_D_R8 (.ARRAY, .ROW, .COLUMN);
3
3        [DSC$K_DTYPE_T] :
3
3            IF .FLAGS AND V_1D_FLAG
3            THEN
3                BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .COLUMN)
3            ELSE
3                BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .ROW, .COLUMN);
3
3        [DSC$K_DTYPE_DSC] :
3
4            BEGIN
4            CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
4                SET
4
4                [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F,
4                 DSC$K_DTYPE_D, DSC$K_DTYPE_G, DSC$K_DTYPE_H, DSC$K_DTYPE_P] :
5                    BEGIN
5
5                    TEMP_STORE [0] = %X'00000000';
5                    IF .FLAGS AND V_1D_FLAG
5                    THEN
5                        BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .COLUMN)
5                    ELSE
5                        BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .ROW, .COLUMN);
5
4                    END;
4
4                [DSC$K_DTYPE_T] :
4
4                    IF .FLAGS AND V_1D_FLAG
4                    THEN
4                        BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .COLUMN)
4                    ELSE
4                        BAS$FETCH_BFA (.ARRAY, TEMP_STORE [0], .ROW, .COLUMN);
4
4                [INRANGE,OUTRANGE] :
4                    BAS$$STOP (BAS$K_DATTYPERR);
4
4                TES;
4
3            END;                               ! end of dtype dsc
3
3        [DSC$K_DTYPE_P] :
3        !+
3        ! Must pass a descriptor to BAS$$UDF_WL1.  Construct a class SD
3        ! descriptor here, and set the pointer field to TEMP_STORE.
3        !-
4            BEGIN
4            NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
```

```
4                   NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
4                   NUM_DESCRIP [DSC$W_LENGTH] = .ARRAY [DSC$W_LENGTH];
4                   NUM_DESCRIP [DSC$B_SCALE] = .ARRAY [DSC$B_SCALE];
4                   NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
4
4                   IF .FLAGS AND V_1D_FLAG
4                   THEN
4                       BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .COLUMN)
4                   ELSE
4                       BAS$FETCH_BFA (.ARRAY, NUM_DESCRIP, .ROW, .COLUMN);
3                   END;
3
3
3               [DSC$K_DTYPE_G] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       TEMP_STORE [0] = BAS$FET_FA_G_R8 (.ARRAY, .COLUMN)
3                   ELSE
3                       TEMP_STORE [0] = BAS$FET_FA_G_R8 (.ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_H] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       TEMP_STORE [0] = BAS$FET_FA_H_R8 (.ARRAY, .COLUMN)
3                   ELSE
3                       TEMP_STORE [0] = BAS$FET_FA_H_R8 (.ARRAY, .ROW, .COLUMN);
3
3               [INRANGE, OUTRANGE] :
3                   BAS$$STOP (BAS$K_DATTYPERR);
3               TES;
3
3           BAS$$UDF_WL1 (
4               BEGIN
4
4               IF (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC) THEN .ELEM_DESCRIP [DSC$B_DTYPE] ELSE .ARRAY [DSC$
4
4               END
3           ,                                         !
4               BEGIN
4               MAP
4                   TEMP_STORE : BLOCK [8,BYTE];
4
5               (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T
5                THEN
5                   .TEMP_STORE [DSC$W_LENGTH]
5                ELSE
6                   (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC
6                    THEN
6                       IF .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T
6                       THEN
6                           .TEMP_STORE [DSC$W_LENGTH]
6                       ELSE
6                           .NUM_DESCRIP [DSC$W_LENGTH]
6                    ELSE
5                       .ARRAY [DSC$W_LENGTH]))
; 1603          2520  4                 END
; 1604          2521  3                                                     !
; 1605          2522  4                 (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
; 1606          2523  4                  THEN
; 1607          2524  4                     NUM_DESCRIP                     ! pass dsc for packed
; 1608          2525  4                  ELSE
; 1609          2526  5                     (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
; 1610          2527  5                         .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
; 1611          2528  5                      THEN
; 1612          2529  5                         NUM_DESCRIP
; 1613          2530  5                      ELSE
; 1614          2531  3                         TEMP_STORE)),
; 1615          2532  3
; 1616          2533  3                 BAS$K_NO_FORM);
; 1617          2534  3             NUM_ELEMS_DONE = .NUM_ELEMS_DONE + 1;
; 1618          2535  3             COLUMN = .COLUMN + 1;
; 1619          2536  3
; 1620          2537  3             IF .COLUMN GTR .UPPER_BOUND1
; 1621          2538  3             THEN
; 1622          2539  4                 BEGIN
; 1623          2540  4 !+
; 1624          2541  4 ! It is time to start a new row.
; 1625          2542  4 !-
; 1626          2543  4                 ROW = .ROW + 1;
; 1627          2544  4                 COLUMN = 1;
; 1628          2545  3                 END;
; 1629          2546  3
; 1630          2547  2             END;                                    ! end of the WHILE loop
; 1631          2548  2
; 1632          2549  2 !+
; 1633          2550  2 ! Return any temporary storage used and then return
; 1634          2551  2 !-
; 1635          2552  2
; 1636          2553  2         IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T OR
; 1637          2554  3           (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
; 1638          2555  3            .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T)
; 1639          2556  2         THEN
; 1640          2557  2             STR$FREE1_DX (TEMP_STORE);
; 1641          2558  2
; 1642          2559  2         RETURN;
; 1643          2560  1         END;                                        !End of BAS$OUT_MAT_B



                                          0FFC 00000          .ENTRY  BAS$OUT_MAT_B, Save R2,R3,R4,R5,R6,R7,R8,-   ; 2121
                                                                      R9,R10,R11                                   ;
                         5E          30   C2 00002          SUBL2   #48, SP                                      ;
                            00000000G 00  16 00005          JSB     BAS$$CB_GET                                  ; 2191
           07        97  AB          02   E1 0000B          BBC     #2, -105(CCB), 1$                            ; 2197
             00000000G 00            00   FB 00010          CALLS   #0, BAS$$BLNK_LINE                           ;
                     97  AB          04   88 00017 1$:      BISB2   #4, -105(CCB)                                ; 2199
                                 08  AE   D4 0001B          CLRL    FLAGS                                        ; 2200
                 20  AE 020E0000     8F   D0 0001E          MOVL    #34471936, TEMP_STORE                        ; 2204
                                 24  AE   D4 00026          CLRL    TEMP_STORE+4                                 ; 2205
                         59      04  AC   D0 00029          MOVL    ARRAY, R9                                    ; 2210
```

```
                                                   50  D4 0002D           CLRL    R0                                          ;
                              01          0B       A9  91 0002F           CMPB    11(R9), #1                                  ;
                                                   05  12 00033           BNEQ    2$                                          ;
                                                   50  D6 00035           INCL    R0                                          ;
                                          08       AE  D6 00037           INCL    FLAGS                                       ;
                              02                   6C  91 0003A  2$:      CMPB    (AP), #2                                    ; 2218
                                                   1B  1E 0003D           BGEQU   4$                                          ;
                              0C                   50  E9 0003F           BLBC    R0, 3$                                      ; 2221
                         0C   AE          1C       A9  D0 00042           MOVL    28(R9), UPPER_BOUND1                        ; 2227
                         14   AE          0C       AE  D0 00047           MOVL    UPPER_BOUND1, TOTAL_NUM_ITEMS               ; 2228
                                                   0C  11 0004C           BRB     4$                                          ; 2221
                         0C   AE          28       A9  D0 0004E  3$:      MOVL    40(R9), UPPER_BOUND1                        ; 2235
         14    AE        20   A9          0C       AE  C5 00053           MULL3   UPPER_BOUND1, 32(R9), TOTAL_NUM_ITEMS       ; 2236
                              02                   6C  91 0005A  4$:      CMPB    (AP), #2                                    ; 2239
                                                   0A  1F 0005D           BLSSU   5$                                          ;
                         0C   AE          08       AC  D0 0005F           MOVL    SUBSCRIPT1, UPPER_BOUND1                    ; 2242
                         14   AE          08       AC  D0 00064           MOVL    SUBSCRIPT1, TOTAL_NUM_ITEMS                 ; 2243
                              03                   6C  91 00069  5$:      CMPB    (AP), #3                                    ; 2246
                                                   0C  12 0006C           BNEQ    6$                                          ;
                         0C   AE          0C       AC  D0 0006E           MOVL    SUBSCRIPT2, UPPER_BOUND1                    ; 2252
         14    AE        08   AC          0C       AC  C5 00073           MULL3   SUBSCRIPT2, SUBSCRIPT1, TOTAL_NUM_ITEMS     ; 2253
                         10   AE                   01  D0 0007A  6$:      MOVL    #1, NUM_ELEMS_DONE                          ; 2259
                              5A                   01  D0 0007E           MOVL    #1, COLUMN                                  ;
                         04   AE                   01  D0 00081           MOVL    #1, ROW                                     ;
                              18          02       A9  91 00085           CMPB    2(R9), #24                                  ; 2267
                                                   7B  12 00089           BNEQ    13$                                         ;
                         1C   AE          20       AE  9E 0008B           MOVAB   TEMP_STORE, NUM_DESCRIP+4                   ; 2271
                              0D          08       AE  E9 00090           BLBC    FLAGS, 7$                                   ; 2272
                                                   01  DD 00094           PUSHL   #1                                          ; 2274
                                                   59  DD 00096           PUSHL   R9                                          ;
                   00000000G  00                   02  FB 00098           CALLS   #2, BAS$FETCH_DESC                          ;
                                                   0D  11 0009F           BRB     8$                                          ;
                                                   01  DD 000A1  7$:      PUSHL   #1                                          ; 2276
                                                   01  DD 000A3           PUSHL   #1                                          ;
                                                   59  DD 000A5           PUSHL   R9                                          ;
                   00000000G  00                   03  FB 000A7           CALLS   #3, BAS$FETCH_DESC                          ;
                              6E                   50  D0 000AE  8$:      MOVL    R0, ELEM_DESCRIP                            ;
                52            6E                   02  C1 000B1           ADDL3   #2, ELEM_DESCRIP, R2                        ; 2279
                16            06                   62  8F 000B5           CASEB   (R2), #6, #22                               ;
 002E          004F          0045                003B     000B9  9$:      .WORD   11$-9$,-                                    ;
 002E          002E          0063                0057     000C1                   12$-9$,-                                    ;
 002E          002E          002E                0090     000C9                   14$-9$,-                                    ;
 0081          002E          002E                002E     000D1                   10$-9$,-                                    ;
 002E          002E          002E                002E     000D9                   15$-9$,-                                    ;
               0077          006B                002E     000E1                   17$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  22$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  21$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
                                                                                  10$-9$,-                                    ;
```

```
                                                                    10$-9$,-                                  ;
                                                                    10$-9$,-                                  ;
                                                                    10$-9$,-                                  ;
                                                                    18$-9$,-                                  ;
                                                                    20$-9$                                    ;
                        7E      00G  8F   9A 000E7 10$:    MOVZBL   #BAS$K_DATTYPERR, -(SP)                   ; 2352
              00000000G 00           01   FB 000EB         CALLS    #1, BAS$$STOP                             ;
                                     55   11 000F2         BRB      22$                                       ;
                    18  AE 01060001  8F   D0 000F4 11$:    MOVL     #17170433, NUM_DESCRIP                    ; 2291
                                     4B   11 000FC         BRB      22$                                       ; 2279
                    18  AE 01070002  8F   D0 000FE 12$:    MOVL     #17235970, NUM_DESCRIP                    ; 2299
                                     41   11 00106 13$:    BRB      22$                                       ; 2279
                    1A  AE     0108  8F   B0 00108 14$:    MOVW     #264, NUM_DESCRIP+2                       ; 2306
                                     06   11 0010E         BRB      16$                                       ; 2307
                    1A  AE     010A  8F   B0 00110 15$:    MOVW     #266, NUM_DESCRIP+2                       ; 2314
                    18  AE           04   B0 00116 16$:    MOVW     #4, NUM_DESCRIP                           ; 2315
                                     2D   11 0011A         BRB      22$                                       ; 2279
                    1A  AE     010B  8F   B0 0011C 17$:    MOVW     #267, NUM_DESCRIP+2                       ; 2322
                                     06   11 00122         BRB      19$                                       ; 2323
                    1A  AE     011B  8F   B0 00124 18$:    MOVW     #283, NUM_DESCRIP+2                       ; 2330
                    18  AE           08   B0 0012A 19$:    MOVW     #8, NUM_DESCRIP                           ; 2331
                                     19   11 0012E         BRB      22$                                       ; 2279
                    18  AE 011C0010  8F   D0 00130 20$:    MOVL     #18612240, NUM_DESCRIP                    ; 2339
                                     0F   11 00138         BRB      22$                                       ; 2279
                    1A  AE     0915  8F   B0 0013A 21$:    MOVW     #2325, NUM_DESCRIP+2                      ; 2346
                    18  AE           69   B0 00140         MOVW     (R9), NUM_DESCRIP                         ; 2347
                    20  AE       08  A9   90 00144         MOVB     8(R9), NUM_DESCRIP+8                      ; 2348
                    14  AE       10  AE   D1 00149 22$:    CMPL     NUM_ELEMS_DONE, TOTAL_NUM_ITEMS           ; 2363
                                     03   15 0014E         BLEQ     23$                                       ;
                                   020B   31 00150         BRW      72$                                       ;
                   16       06       02  A9   8F 00153 23$:    CASEB    2(R9), #6, #22                        ; 2371
   00FA          0067     004C     0031      00158 24$:    .WORD    25$-24$,-                                 ;
   00FA          00FA     009D     0082      00160                  28$-24$,-                                 ;
   00FA          00FA     00FA     00B9      00168                  31$-24$,-                                 ;
   0116          00FA     00FA     00FA      00170                  44$-24$,-                                 ;
   00FA          00C4     00FA     00FA      00178                  34$-24$,-                                 ;
                 016C     0151     00FA      00180                  37$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    41$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    47$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    42$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    44$-24$,-                                 ;
                                                                    53$-24$,-                                 ;
                                                                    56$-24$                                   ;
                                   00C9   31 00186         BRW      44$                                       ; 2493
                        05       08  AE   E9 00189 25$:    BLBC     FLAGS, 26$                                ; 2376
                        51           5A   D0 0018D         MOVL     COLUMN, R1                                ; 2378
```

```
                                       07  11 00190       BRB     27$                                  ;
                      52               5A  D0 00192 26$:  MOVL    COLUMN, R2                           ; 2380
                      51         04    AE  D0 00195       MOVL    ROW, R1                              ;
                      50               59  D0 00199 27$:  MOVL    R9, R0                               ;
                          00000000G    00  16 0019C       JSB     BAS$FET_FA_B_R8                      ;
                                       6A  11 001A2       BRB     40$                                  ;
                      05         08    AE  E9 001A4 28$:  BLBC    FLAGS, 29$                           ; 2384
                      51               5A  D0 001A8       MOVL    COLUMN, R1                           ; 2386
                                       07  11 001AB       BRB     30$                                  ;
                      52               5A  D0 001AD 29$:  MOVL    COLUMN, R2                           ; 2388
                      51         04    AE  D0 001B0       MOVL    ROW, R1                              ;
                      50               59  D0 001B4 30$:  MOVL    R9, R0                               ;
                          00000000G    00  16 001B7       JSB     BAS$FET_FA_W_R8                      ;
                                       4F  11 001BD       BRB     40$                                  ;
                      05         08    AE  E9 001BF 31$:  BLBC    FLAGS, 32$                           ; 2392
                      51               5A  D0 001C3       MOVL    COLUMN, R1                           ; 2394
                                       07  11 001C6       BRB     33$                                  ;
                      52               5A  D0 001C8 32$:  MOVL    COLUMN, R2                           ; 2396
                      51         04    AE  D0 001CB       MOVL    ROW, R1                              ;
                      50               59  D0 001CF 33$:  MOVL    R9, R0                               ;
                          00000000G    00  16 001D2       JSB     BAS$FET_FA_L_R8                      ;
                                       34  11 001D8       BRB     40$                                  ;
                      05         08    AE  E9 001DA 34$:  BLBC    FLAGS, 35$                           ; 2400
                      51               5A  D0 001DE       MOVL    COLUMN, R1                           ; 2402
                                       07  11 001E1       BRB     36$                                  ;
                      52               5A  D0 001E3 35$:  MOVL    COLUMN, R2                           ; 2404
                      51         04    AE  D0 001E6       MOVL    ROW, R1                              ;
                      50               59  D0 001EA 36$:  MOVL    R9, R0                               ;
                          00000000G    00  16 001ED       JSB     BAS$FET_FA_F_R8                      ;
                                       19  11 001F3       BRB     40$                                  ;
                      05         08    AE  E9 001F5 37$:  BLBC    FLAGS, 38$                           ; 2408
                      51               5A  D0 001F9       MOVL    COLUMN, R1                           ; 2410
                                       07  11 001FC       BRB     39$                                  ;
                      52               5A  D0 001FE 38$:  MOVL    COLUMN, R2                           ; 2412
                      51         04    AE  D0 00201       MOVL    ROW, R1                              ;
                      50               59  D0 00205 39$:  MOVL    R9, R0                               ;
                          00000000G    00  16 00208       JSB     BAS$FET_FA_D_R8                      ;
                                     00CC  31 0020E 40$:  BRW     59$                                  ;
                      4F         08    AE  E9 00211 41$:  BLBC    FLAGS, 46$                           ; 2416
                                       5A  DD 00215       PUSHL   COLUMN                               ; 2418
                                 24    AE  9F 00217       PUSHAB  TEMP_STORE                           ;
                                       6F  11 0021A       BRB     49$                                  ;
               52     6E               02  C1 0021C 42$:  ADDL3   #2, ELEM_DESCRIP, R2                 ; 2425
               16     06               62  8F 00220       CASEB   (R2), #6, #22                        ;
   002E      003B   003B             003B     00224 43$:  .WORD   45$-43$,-                            ;
   002E      002E   003B             003B     0022C               45$-43$,-                            ;
   002E      002E   002E             FFED     00234               45$-43$,-                            ;
   003B      002E   002E             002E     0023C               44$-43$,-                            ;
   002E      002E   002E             002E     00244               45$-43$,-                            ;
             003B   003B             002E     0024C               45$-43$,-                            ;
                                                                  44$-43$,-                            ;
                                                                  44$-43$,-                            ;
                                                                  41$-43$,-                            ;
                                                                  44$-43$,-                            ;
                                                                  44$-43$,-                            ;
                                                                  44$-43$,-                            ;
                                                                  44$-43$,-                            ;
                                                                  44$-43$,-                                    ;
                                                                  44$-43$,-                                    ;
                                                                  45$-43$,-                                    ;
                                                                  44$-43$,-                                    ;
                                                                  44$-43$,-                                    ;
                                                                  44$-43$,-                                    ;
                                                                  44$-43$,-                                    ;
                                                                  44$-43$,-                                    ;
                                                                  45$-43$,-                                    ;
                                                                  45$-43$                                      ;
               7E        00G  8F    9A 00252 44$:     MOVZBL   #BAS$K_DATTYPERR, -(SP)                        ; 2450
    00000000G  00              01   FB 00256          CALLS    #1, BAS$$STOP                                  ;
                               48   11 0025D          BRB      52$                                            ;
                         20   AE    D4 0025F 45$:     CLRL     TEMP_STORE                                     ; 2432
                               1E   11 00262          BRB      48$                                            ; 2433
                               5A   DD 00264 46$:     PUSHL    COLUMN                                         ; 2447
                         08   AE    DD 00266          PUSHL    ROW                                            ;
                         28   AE    9F 00269          PUSHAB   TEMP_STORE                                     ;
                               30   11 0026C          BRB      51$                                            ;
          1A   AE      0915   8F    B0 0026E 47$:     MOVW     #2325, NUM_DESCRIP+2                           ; 2463
          18   AE              69   B0 00274          MOVW     (R9), NUM_DESCRIP                              ; 2464
          20   AE        08   A9    90 00278          MOVB     8(R9), NUM_DESCRIP+8                           ; 2465
          1C   AE        20   AE    9E 0027D          MOVAB    TEMP_STORE, NUM_DESCRIP+4                      ; 2466
               10        08   AE    E9 00282 48$:     BLBC     FLAGS, 50$                                     ; 2468
                               5A   DD 00286          PUSHL    COLUMN                                         ; 2470
                         1C   AE    9F 00288          PUSHAB   NUM_DESCRIP                                    ;
                               59   DD 0028B 49$:     PUSHL    R9                                             ;
    00000000G  00              03   FB 0028D          CALLS    #3, BAS$FETCH_BFA                              ;
                               4B   11 00294          BRB      60$                                            ;
                               5A   DD 00296 50$:     PUSHL    COLUMN                                         ; 2472
                         08   AE    DD 00298          PUSHL    ROW                                            ;
                         20   AE    9F 0029B          PUSHAB   NUM_DESCRIP                                    ;
                               59   DD 0029E 51$:     PUSHL    R9                                             ;
    00000000G  00              04   FB 002A0          CALLS    #4, BAS$FETCH_BFA                              ;
                               38   11 002A7 52$:     BRB      60$                                            ; 2371
               05        08   AE    E9 002A9 53$:     BLBC     FLAGS, 54$                                     ; 2478
               51              5A   D0 002AD          MOVL     COLUMN, R1                                     ; 2480
                               07   11 002B0          BRB      55$                                            ;
               52              5A   D0 002B2 54$:     MOVL     COLUMN, R2                                     ; 2482
               51        04   AE    D0 002B5          MOVL     ROW, R1                                        ;
               50              59   D0 002B9 55$:     MOVL     R9, R0                                         ;
                    00000000G  00   16 002BC          JSB      BAS$FET_FA_G_R8                                ;
                               19   11 002C2          BRB      59$                                            ;
               05        08   AE    E9 002C4 56$:     BLBC     FLAGS, 57$                                     ; 2486
               51              5A   D0 002C8          MOVL     COLUMN, R1                                     ; 2488
                               07   11 002CB          BRB      58$                                            ;
               52              5A   D0 002CD 57$:     MOVL     COLUMN, R2                                     ; 2490
               51        04   AE    D0 002D0          MOVL     ROW, R1                                        ;
               50              59   D0 002D4 58$:     MOVL     R9, R0                                         ;
                    00000000G  00   16 002D7          JSB      BAS$FET_FA_H_R8                                ;
          20   AE              50   D0 002DD 59$:     MOVL     R0, TEMP_STORE                                 ;
                               03   DD 002E1 60$:     PUSHL    #3                                             ; 2496
               50        02   A9    9A 002E3          MOVZBL   2(R9), R0                                      ; 2522
               15              50   91 002E7          CMPB     R0, #21                                        ;
                               0F   13 002EA          BEQL     61$                                            ;
               18              50   91 002EC          CMPB     R0, #24                                        ; 2526
                               10   12 002EF          BNEQ     62$                                            ;
          51      04  AE          02  C1 002F1          ADDL3   #2, ELEM_DESCRIP, R1                  ; 2527
                      15          61  91 002F6          CMPB    (R1), #21                             ;
                      06          12  002F9             BNEQ    62$                                   ;
                      51      1C  AE  9E 002FB 61$:     MOVAB   NUM_DESCRIP, R1                       ; 2526
                      04          11  002FF             BRB     63$                                   ;
                      51      24  AE  9E 00301 62$:     MOVAB   TEMP_STORE, R1                        ;
                                  51  DD 00305 63$:     PUSHL   R1                                    ;
                      0E          50  91 00307          CMPB    R0, #14                               ; 2507
                                  06  12 0030A          BNEQ    64$                                   ;
                      7E      28  AE  3C 0030C          MOVZWL  TEMP_STORE, -(SP)                     ; 2509
                                  20  11 00310          BRB     68$                                   ;
                      18          50  91 00312 64$:     CMPB    R0, #24                               ; 2511
                                  16  12 00315          BNEQ    66$                                   ;
          51      08  AE          02  C1 00317          ADDL3   #2, ELEM_DESCRIP, R1                  ; 2513
                      0E          61  91 0031C          CMPB    (R1), #14                             ;
                                  06  12 0031F          BNEQ    65$                                   ;
                      51      28  AE  3C 00321          MOVZWL  TEMP_STORE, R1                        ; 2515
                                  09  11 00325          BRB     67$                                   ;
                      51      20  AE  3C 00327 65$:     MOVZWL  NUM_DESCRIP, R1                       ; 2517
                                  03  11 0032B          BRB     67$                                   ; 2513
                      51          69  3C 0032D 66$:     MOVZWL  (R9), R1                              ; 2519
                                  51  DD 00330 67$:     PUSHL   R1                                    ; 2511
                      18          50  91 00332 68$:     CMPB    R0, #24                               ; 2499
                                  0A  12 00335          BNEQ    69$                                   ;
          51      0C  AE          02  C1 00337          ADDL3   #2, ELEM_DESCRIP, R1                  ;
                      7E          61  9A 0033C          MOVZBL  (R1), -(SP)                           ;
                                  02  11 0033F          BRB     70$                                   ;
                                  50  DD 00341 69$:     PUSHL   R0                                    ;
           00000000G  00          04  FB 00343 70$:     CALLS   #4, BAS$$UDF_WL1                      ; 2497
                              10  AE  D6 0034A          INCL    NUM_ELEMS_DONE                        ; 2534
                                  5A  D6 0034D          INCL    COLUMN                                ; 2535
                  0C  AE          5A  D1 0034F          CMPL    COLUMN, UPPER_BOUND1                  ; 2537
                                  06  15 00353          BLEQ    71$                                   ;
                              04  AE  D6 00355          INCL    ROW                                   ; 2543
                      5A          01  D0 00358          MOVL    #1, COLUMN                            ; 2544
                            FDEB  31 0035B 71$:         BRW     22$                                   ; 2363
                      0E      02  A9  91 0035E 72$:     CMPB    2(R9), #14                            ; 2553
                                  0F  13 00362          BEQL    73$                                   ;
                      18      02  A9  91 00364          CMPB    2(R9), #24                            ; 2554
                                  13  12 00368          BNEQ    74$                                   ;
          50          6E          02  C1 0036A          ADDL3   #2, ELEM_DESCRIP, R0                  ; 2555
                      0E          60  91 0036E          CMPB    (R0), #14                             ;
                                  0A  12 00371          BNEQ    74$                                   ;
                              20  AE  9F 00373 73$:     PUSHAB  TEMP_STORE                            ; 2557
           00000000G  00          01  FB 00376          CALLS   #1, STR$FREE1_DX                      ;
                                  04  0037D 74$:        RET                                           ; 2560

; Routine Size:  894 bytes,    Routine Base:  _BAS$CODE + 070C


; 1644          2561  1
```

```
1 GLOBAL ROUTINE BAS$IN_MAT (                          ! Matrix input
1               ARRAY                                  ! array to print
1       ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !             The array is input one element at a time by rows.  Input may be con-
1 !             tinued on the next line by an '&'.  Only those elements for which new
1 !             data is entered are changed.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !             ARRAY.wx.a                      The array to put the data into
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !             NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !             NUM                     number of rows or elements entered
1 !             NUM2                    the number of elements entered in the last row
1 !                                     if two dimensional
1 !
1 ! COMPLETION CODES:
1 !
1 !             NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !             Signals:
1 !                 Invalid data type
1 !
1 !--
1
2       BEGIN
2
2       GLOBAL REGISTER
2           CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2       LITERAL
2           V_1D_FLAG = 1,                          ! flag - one dimen. array
2           K_1D = 1;                               ! one dimension
2
2       LOCAL
2           NUM_ELEMS_DONE,                         ! total number of array elements processed
2           FLAGS,
2           TEMP_STORE : VECTOR [4, LONG],          ! temp storage for calling FETCH_VA
2           ROW,                                    ! current value of subscript 1
2           COLUMN,                                 ! current value of subscript 2
2           UPPER_BOUND1,                           ! upper bound for 1 dimensional
2                                                   ! array and number of rows for 2
2                                                   ! dimensional array
2           TOTAL_NUM_ITEMS,                        ! total number of items in the array
2                                                   ! excluding row and col. 0
2           ELEM_DESCRIP : REF BLOCK [12,BYTE],     ! desc fetched from array
```

```
2          NUM_DESCRIP : BLOCK [8,BYTE];          ! temp numeric desc for STORE
2
2      MAP
2          ARRAY : REF BLOCK [, BYTE];
2
2      BAS$$CB_GET ();
2      FLAGS = 0;
2  !+
2  ! Default TEMP_STORE to a dynamic stirng descriptor
2  !-
2      TEMP_STORE [0] = %X'020E0000';
2      TEMP_STORE [1] = %X'00000000';
2  !+
2  ! Check number of dimensions and initialize the number of elements in the array.
2  ! Set a flag if only one dimension.
2  !-
2
2      IF .ARRAY [DSC$B_DIMCT] EQL K_1D
2      THEN
3          BEGIN
3          FLAGS = .FLAGS + V_1D_FLAG;
3          UPPER_BOUND1 = .ARRAY [U1_1D];
3          TOTAL_NUM_ITEMS = .UPPER_BOUND1;
3          END
2      ELSE
3          BEGIN
3          UPPER_BOUND1 = .ARRAY [U2_2D];
3          TOTAL_NUM_ITEMS = .ARRAY [U1_2D]*.UPPER_BOUND1;
2          END;
2
2  !+
2  ! Initialize the two current subscripts regardless of the number of dimensions
2  !-
2      ROW = COLUMN = NUM_ELEMS_DONE = 1;
2
2  !+
2  ! If this is an array of descriptors, they may be dynamic string descriptors or
2  ! numeric descriptors in the case of a dynamically mapped array.  Check the
2  ! first element descriptor to determine the dtype (all elements of the array
2  ! should be the same).
2  !-
2
2      IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC
2      THEN
3      BEGIN
3          IF .FLAGS AND V_1D_FLAG
3          THEN
3              ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1)
3          ELSE
3              ELEM_DESCRIP = BAS$FETCH_DESC (.ARRAY, 1, 1);
3
3      CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
3          SET
3
3          [DSC$K_DTYPE_B] :
4              BEGIN
4              NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
```

```
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_B;
        NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/4;
        END;

    [DSC$K_DTYPE_W] :
        BEGIN
        NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_W;
        NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL/2;
        END;

    [DSC$K_DTYPE_L] :
        BEGIN
        NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_L;
        NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
        END;

    [DSC$K_DTYPE_F] :
        BEGIN
        NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_F;
        NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL;
        END;

    [DSC$K_DTYPE_D] :
        BEGIN
        NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_D;
        NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
        END;

    [DSC$K_DTYPE_T] :
        ;

    [DSC$K_DTYPE_P] :
        BEGIN
        NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
        NUM_DESCRIP [DSC$W_LENGTH] = .ELEM_DESCRIP [DSC$W_LENGTH];
        NUM_DESCRIP [DSC$B_SCALE] = .ELEM_DESCRIP [DSC$B_SCALE];
        END;

    [DSC$K_DTYPE_G] :
        BEGIN
        NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_G;
        NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*2;
        END;

    [DSC$K_DTYPE_H] :
        BEGIN
        NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_S;
        NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_H;
        NUM_DESCRIP [DSC$W_LENGTH] = %UPVAL*4;
        END;
```

```
3            [INRANGE,OUTRANGE] :
3                BAS$$STOP (BAS$K_DATTYPERR);
3
3            TES;
3
3        NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
3
2        END;                                    ! dtype dsc
2
2        IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
2        THEN
3            BEGIN
3            NUM_DESCRIP [DSC$B_CLASS] = DSC$K_CLASS_SD;
3            NUM_DESCRIP [DSC$B_DTYPE] = DSC$K_DTYPE_P;
3            NUM_DESCRIP [DSC$W_LENGTH] = .ARRAY [DSC$W_LENGTH];
3            NUM_DESCRIP [DSC$B_SCALE] = .ARRAY [DSC$B_SCALE];
3            NUM_DESCRIP [DSC$A_POINTER] = TEMP_STORE [0];
2            END;
2
2   !+
2   ! Loop thru the array descriptor until all of the elements in the array or as
2   ! many as are supplied are input.
2   !-
2
2        WHILE (.NUM_ELEMS_DONE LEQ .TOTAL_NUM_ITEMS) AND
3            (BAS$$UDF_RL1 (
4                (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC THEN .ELEM_DESCRIP [DSC$B_DTYPE]
3                 ELSE .ARRAY [DSC$B_DTYPE]),            !
5                (IF (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T)
4                 THEN
4                    .TEMP_STORE [0]
4                 ELSE
5                    (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC
5                    THEN
5                        IF .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T
5                        THEN
5                            .TEMP_STORE [0]
5                        ELSE
5                            .NUM_DESCRIP [DSC$W_LENGTH]
5                    ELSE
3                        .ARRAY [DSC$W_LENGTH])),       !
4                (IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_P OR
5                   (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
5                    .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P)
4                 THEN
4                    NUM_DESCRIP                        ! pass desc for packed
4                 ELSE
3                    TEMP_STORE),                       !
2                BAS$K_NULL)) DO
3            BEGIN
3   !+
3   ! Based on the data type, JSB or CALL the proper store routine to put the element
3   ! into the array.  The FETCH and STORE routines are called because the array
3   ! may be virtual.
3   !-
3
3            IF .COLUMN GTR .UPPER_BOUND1
```

```
3           THEN
4               BEGIN
4   !+
4   ! It is time to start a new row.
4   !-
4               ROW = .ROW + 1;
4               COLUMN = 1;
3               END;
3
3           CASE .ARRAY [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
3               SET
3
3               [DSC$K_DTYPE_B] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       BAS$STO_FA_B_R8 (.TEMP_STORE [0], .ARRAY, .COLUMN)
3                   ELSE
3                       BAS$STO_FA_B_R8 (.TEMP_STORE [0], .ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_W] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       BAS$STO_FA_W_R8 (.TEMP_STORE [0], .ARRAY, .COLUMN)
3                   ELSE
3                       BAS$STO_FA_W_R8 (.TEMP_STORE [0], .ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_L] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       BAS$STO_FA_L_R8 (.TEMP_STORE [0], .ARRAY, .COLUMN)
3                   ELSE
3                       BAS$STO_FA_L_R8 (.TEMP_STORE [0], .ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_F] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       BAS$STO_FA_F_R8 (.TEMP_STORE [0], .ARRAY, .COLUMN)
3                   ELSE
3                       BAS$STO_FA_F_R8 (.TEMP_STORE [0], .ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_D] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
3                       BAS$STO_FA_D_R8 (.TEMP_STORE [0], .TEMP_STORE [1], .ARRAY,
3                           .COLUMN)
3                   ELSE
3                       BAS$STO_FA_D_R8 (.TEMP_STORE [0], .TEMP_STORE [1], .ARRAY, .ROW, .COLUMN);
3
3               [DSC$K_DTYPE_T] :
3
3                   IF .FLAGS AND V_1D_FLAG
3                   THEN
```

```
                BAS$STORE_BFA (TEMP_STORE [0], .ARRAY, .COLUMN)
            ELSE
                BAS$STORE_BFA (TEMP_STORE [0], .ARRAY, .ROW, .COLUMN);

        [DSC$K_DTYPE_DSC] :

            BEGIN
            CASE .ELEM_DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
                SET

                [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F,
                 DSC$K_DTYPE_D, DSC$K_DTYPE_G, DSC$K_DTYPE_H, DSC$K_DTYPE_P] :

                    IF .FLAGS AND V_1D_FLAG
                    THEN
                        BAS$STORE_BFA (NUM_DESCRIP, .ARRAY, .COLUMN)
                    ELSE
                        BAS$STORE_BFA (NUM_DESCRIP, .ARRAY, .ROW, .COLUMN);

                [DSC$K_DTYPE_T] :

                    IF .FLAGS AND V_1D_FLAG
                    THEN
                        BAS$STORE_BFA (TEMP_STORE [0], .ARRAY, .COLUMN)
                    ELSE
                        BAS$STORE_BFA (TEMP_STORE [0], .ARRAY, .ROW, .COLUMN);

                [INRANGE,OUTRANGE] :

                        BAS$$STOP (BAS$K_DATTYPERR);

                    TES;

                END;                                    ! data type dsc

            [DSC$K_DTYPE_P] :

                IF .FLAGS AND V_1D_FLAG
                THEN
                    BAS$STORE_BFA (NUM_DESCRIP, .ARRAY, .COLUMN)
                ELSE
                    BAS$STORE_BFA (NUM_DESCRIP, .ARRAY, .ROW, .COLUMN);

            [DSC$K_DTYPE_G] :

                IF .FLAGS AND V_1D_FLAG
                THEN
                    BAS$STO_FA_G_R8 (.TEMP_STORE [0], .TEMP_STORE [1], .ARRAY, .COLUMN)
                ELSE
                    BAS$STO_FA_G_R8 (.TEMP_STORE [0], .TEMP_STORE [1], .ARRAY, .ROW, .COLUMN);

            [DSC$K_DTYPE_H] :

                IF .FLAGS AND V_1D_FLAG
                THEN
                    BAS$STO_FA_H_R8 (.TEMP_STORE [0], .TEMP_STORE [1],
                                     .TEMP_STORE [2], .TEMP_STORE [3], .ARRAY, .COLUMN)
                ELSE
                    BAS$STO_FA_H_R8 (.TEMP_STORE [0], .TEMP_STORE [1],
                                     .TEMP_STORE [2], .TEMP_STORE [3], .ARRAY, .ROW, .COLUMN);

            [INRANGE, OUTRANGE] :
                BAS$$STOP (BAS$K_DATTYPERR);
            TES;

        NUM_ELEMS_DONE = .NUM_ELEMS_DONE + 1;
        COLUMN = .COLUMN + 1;
        END;                                            ! end of the WHILE loop

    NUM = (IF .FLAGS AND V_1D_FLAG THEN .COLUMN - 1 ELSE .ROW);
    NUM2 = (IF .FLAGS AND V_1D_FLAG THEN 0 ELSE .COLUMN - 1);
!+
! Return any temporary storage used and then return
!-

    IF .ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_T OR
      (.ARRAY [DSC$B_DTYPE] EQL DSC$K_DTYPE_DSC AND
       .ELEM_DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_T)
    THEN
        STR$FREE1_DX (TEMP_STORE);

    RETURN;
    END;                                                !End of BAS$IN_MAT
```

```
                                      0FFC 00000          .ENTRY  BAS$IN_MAT, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 2562
                                                                  R10,R11                                    ;
                      5E              34   C2 00002       SUBL2   #52, SP                                    ;
                           00000000G  00   16 00005       JSB     BAS$$CB_GET                                ; 2624
                                      6E   D4 0000B       CLRL    FLAGS                                      ; 2625
                  24  AE  020E0000    8F   D0 0000D       MOVL    #34471936, TEMP_STORE                      ; 2629
                                28    AE   D4 00015       CLRL    TEMP_STORE+4                               ; 2630
                  59            04    AC   D0 00018       MOVL    ARRAY, R9                                  ; 2636
                  01            0B    A9   91 0001C       CMPB    11(R9), #1                                 ;
                                      0E   12 00020       BNEQ    1$                                         ;
                                      6E   D6 00022       INCL    FLAGS                                      ; 2639
                  10  AE        1C    A9   D0 00024       MOVL    28(R9), UPPER_BOUND1                       ; 2640
                  14  AE        10    AE   D0 00029       MOVL    UPPER_BOUND1, TOTAL_NUM_ITEMS              ; 2641
                                      0C   11 0002E       BRB     2$                                         ; 2636
                  10  AE        28    A9   D0 00030 1$:   MOVL    40(R9), UPPER_BOUND1                       ; 2645
      14   AE     20  A9        10    AE   C5 00035       MULL3   UPPER_BOUND1, 32(R9), TOTAL_NUM_ITEMS      ; 2646
                  18  AE              01   D0 0003C 2$:   MOVL    #1, NUM_ELEMS_DONE                         ; 2652
                  04  AE              01   D0 00040       MOVL    #1, COLUMN                                 ;
                  08  AE              01   D0 00044       MOVL    #1, ROW                                    ;
                  0C  AE        02    A9   9E 00048       MOVAB   2(R9), 12(SP)                              ; 2661
                      18        0C    BE   91 0004D       CMPB    @12(SP), #24                               ;
                                      03   13 00051       BEQL    3$                                         ;
                                    00BA   31 00053       BRW     19$                                        ;
                      0D              6E   E9 00056 3$:   BLBC    FLAGS, 4$                                  ; 2664
                                      01   DD 00059       PUSHL   #1                                         ; 2666
                                      59   DD 0005B       PUSHL   R9                                         ;
             00000000G  00            02   FB 0005D       CALLS   #2, BAS$FETCH_DESC                         ;
                                      0D   11 00064       BRB     5$                                         ;
                                      01   DD 00066 4$:   PUSHL   #1                                         ; 2668
                                      01   DD 00068       PUSHL   #1                                         ;
                                      59   DD 0006A       PUSHL   R9                                         ;
             00000000G  00            03   FB 0006C       CALLS   #3, BAS$FETCH_DESC                         ;
                        5A            50   D0 00073 5$:   MOVL    R0, ELEM_DESCRIP                           ;
               16       06      02    AA   8F 00076       CASEB   2(ELEM_DESCRIP), #6, #22                   ; 2670
  002E         004F     0045        003B      0007B 6$:   .WORD   8$-6$,-                                    ;
  002E         002E     0063        0057      00083               9$-6$,-                                    ;
  002E         002E     002E        0090      0008B               10$-6$,-                                   ;
  006B         002E     002E        002E      00093               7$-6$,-                                    ;
  002E         002E     002E        002E      0009B               11$-6$,-                                   ;
               0088     007C        002E      000A3               13$-6$,-                                   ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  18$-6$,-                                   ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  14$-6$,-                                   ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
                                                                  7$-6$,-                                    ;
```

```
                                                                 15$-6$,-                                    ;
                                                                 17$-6$                                      ;
                    7E        00G   8F  9A 000A9 7$:    MOVZBL   #BAS$K_DATTYPERR, -(SP)                     ; 2734
          00000000G 00              01  FB 000AD        CALLS    #1, BAS$$STOP                               ;
                                    55  11 000B4        BRB      18$                                         ;
                 1C AE 01060001     8F  D0 000B6 8$:    MOVL     #17170433, NUM_DESCRIP                      ; 2677
                                    4B  11 000BE        BRB      18$                                         ; 2670
                 1C AE 01070002     8F  D0 000C0 9$:    MOVL     #17235970, NUM_DESCRIP                      ; 2684
                                    41  11 000C8        BRB      18$                                         ; 2670
                 1E AE       0108   8F  B0 000CA 10$:   MOVW     #264, NUM_DESCRIP+2                         ; 2690
                                    06  11 000D0        BRB      12$                                         ; 2691
                 1E AE       010A   8F  B0 000D2 11$:   MOVW     #266, NUM_DESCRIP+2                         ; 2697
                 1C AE              04  B0 000D8 12$:   MOVW     #4, NUM_DESCRIP                             ; 2698
                                    2D  11 000DC        BRB      18$                                         ; 2670
                 1E AE       010B   8F  B0 000DE 13$:   MOVW     #267, NUM_DESCRIP+2                         ; 2704
                                    17  11 000E4        BRB      16$                                         ; 2705
                 1E AE       0915   8F  B0 000E6 14$:   MOVW     #2325, NUM_DESCRIP+2                        ; 2714
                 1C AE              6A  B0 000EC        MOVW     (ELEM_DESCRIP), NUM_DESCRIP                 ; 2715
                 24 AE         08   AA  90 000F0        MOVB     8(ELEM_DESCRIP), NUM_DESCRIP+8              ; 2716
                                    14  11 000F5        BRB      18$                                         ; 2670
                 1E AE       011B   8F  B0 000F7 15$:   MOVW     #283, NUM_DESCRIP+2                         ; 2722
                 1C AE              08  B0 000FD 16$:   MOVW     #8, NUM_DESCRIP                             ; 2723
                                    08  11 00101        BRB      18$                                         ; 2670
                 1C AE 011C0010     8F  D0 00103 17$:   MOVL     #18612240, NUM_DESCRIP                      ; 2730
                 20 AE         24   AE  9E 0010B 18$:   MOVAB    TEMP_STORE, NUM_DESCRIP+4                   ; 2738
                    15         0C   BE  91 00110 19$:   CMPB     @12(SP), #21                                ; 2742
                                    14  12 00114        BNEQ     20$                                         ;
                 1E AE       0915   8F  B0 00116        MOVW     #2325, NUM_DESCRIP+2                        ; 2746
                 1C AE              69  B0 0011C        MOVW     (R9), NUM_DESCRIP                           ; 2747
                 24 AE         08   A9  90 00120        MOVB     8(R9), NUM_DESCRIP+8                        ; 2748
                 20 AE         24   AE  9E 00125        MOVAB    TEMP_STORE, NUM_DESCRIP+4                   ; 2749
                 14 AE         18   AE  D1 0012A 20$:   CMPL     NUM_ELEMS_DONE, TOTAL_NUM_ITEMS             ; 2757
                                    03  15 0012F        BLEQ     22$                                         ;
                                  0213  31 00131 21$:   BRW      69$                                         ;
                                    7E  D4 00134 22$:   CLRL     -(SP)                                       ; 2758
                    15         10   BE  91 00136        CMPB     @16(SP), #21                                ; 2774
                                    0C  13 0013A        BEQL     23$                                         ;
                    18         10   BE  91 0013C        CMPB     @16(SP), #24                                ; 2775
                                    0C  12 00140        BNEQ     24$                                         ;
                    15         02   AA  91 00142        CMPB     2(ELEM_DESCRIP), #21                        ; 2776
                                    06  12 00146        BNEQ     24$                                         ;
                    50         20   AE  9E 00148 23$:   MOVAB    NUM_DESCRIP, R0                             ; 2774
                                    04  11 0014C        BRB      25$                                         ;
                    50         28   AE  9E 0014E 24$:   MOVAB    TEMP_STORE, R0                              ;
                                    50  DD 00152 25$:   PUSHL    R0                                          ;
                    0E         14   BE  91 00154        CMPB     @20(SP), #14                                ; 2761
                                    05  12 00158        BNEQ     26$                                         ;
                               2C   AE  DD 0015A        PUSHL    TEMP_STORE                                  ; 2763
                                    1D  11 0015D        BRB      30$                                         ;
                    18         14   BE  91 0015F 26$:   CMPB     @20(SP), #24                                ; 2765
                                    12  12 00163        BNEQ     28$                                         ;
                    0E         02   AA  91 00165        CMPB     2(ELEM_DESCRIP), #14                        ; 2767
                                    06  12 00169        BNEQ     27$                                         ;
                    51         2C   AE  D0 0016B        MOVL     TEMP_STORE, R1                              ; 2769
                                    09  11 0016F        BRB      29$                                         ;
                    51         24   AE  3C 00171 27$:   MOVZWL   NUM_DESCRIP, R1                             ; 2771
                                    03  11 00175        BRB      29$                                         ; 2767
```

```
                                51          69  3C 00177 28$:     MOVZWL  (R9), R1                                ; 2773
                                            51  DD 0017A 29$:     PUSHL   R1                                      ; 2765
                                18    18    BE  91 0017C 30$:     CMPB    @24(SP), #24                            ; 2759
                                            06  12 00180          BNEQ    31$                                     ;
                                7E    02    AA  9A 00182          MOVZBL  2(ELEM_DESCRIP), -(SP)                  ;
                                            04  11 00186          BRB     32$                                     ;
                                7E    18    BE  9A 00188 31$:     MOVZBL  @24(SP), -(SP)                          ; 2760
                     00000000G  00          04  FB 0018C 32$:     CALLS   #4, BAS$$UDF_RL1                        ; 2759
                                9B          50  E9 00193          BLBC    R0, 21$                                 ;
                          10    AE    04    AE  D1 00196          CMPL    COLUMN, UPPER_BOUND1                    ; 2789
                                            07  15 0019B          BLEQ    33$                                     ;
                                      08    AE  D6 0019D          INCL    ROW                                     ; 2795
                          04    AE          01  D0 001A0          MOVL    #1, COLUMN                              ; 2796
                     16         06    0C    BE  8F 001A4 33$:     CASEB   @12(SP), #6, #22                        ; 2799
  0111           0071         0051        0031     001A9 34$:     .WORD   35$-34$,-                               ;
  0111           0111         00B1        0091     001B1                  38$-34$,-                               ;
  0111           0111         00D1        0111     001B9                  41$-34$,-                               ;
  012B           0111         0111        0111     001C1                  54$-34$,-                               ;
  0111           00DE         0111        0111     001C9                  44$-34$,-                               ;
                 0173         0153        0111     001D1                  47$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          51$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          57$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          52$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          54$-34$,-                               ;
                                                                          62$-34$,-                               ;
                                                                          65$-34$                                 ;
                                          00E0  31 001D7          BRW     54$                                     ; 2909
                                06          6E  E9 001DA 35$:     BLBC    FLAGS, 36$                              ; 2804
                                52    04    AE  D0 001DD          MOVL    COLUMN, R2                              ; 2806
                                            08  11 001E1          BRB     37$                                     ;
                                53    04    AE  D0 001E3 36$:     MOVL    COLUMN, R3                              ; 2808
                                52    08    AE  D0 001E7          MOVL    ROW, R2                                 ;
                                51          59  D0 001EB 37$:     MOVL    R9, R1                                  ;
                                50    24    AE  D0 001EE          MOVL    TEMP_STORE, R0                          ;
                           00000000G        00  16 001F2          JSB     BAS$STO_FA_B_R8                         ;
                                            7E  11 001F8          BRB     50$                                     ; 2804
                                06          6E  E9 001FA 38$:     BLBC    FLAGS, 39$                              ; 2812
                                52    04    AE  D0 001FD          MOVL    COLUMN, R2                              ; 2814
                                            08  11 00201          BRB     40$                                     ;
                                53    04    AE  D0 00203 39$:     MOVL    COLUMN, R3                              ; 2816
                                52    08    AE  D0 00207          MOVL    ROW, R2                                 ;
                                51          59  D0 0020B 40$:     MOVL    R9, R1                                  ;
                                50    24    AE  D0 0020E          MOVL    TEMP_STORE, R0                          ;
                           00000000G        00  16 00212          JSB     BAS$STO_FA_W_R8                         ;
                                            5E  11 00218          BRB     50$                                     ; 2812
                                06          6E  E9 0021A 41$:     BLBC    FLAGS, 42$                              ; 2820
```

```
                              52        04  AE  D0 0021D          MOVL    COLUMN, R2                                  ; 2822
                                            08  11 00221          BRB     43$                                         ;
                              53        04  AE  D0 00223 42$:     MOVL    COLUMN, R3                                  ; 2824
                              52        08  AE  D0 00227          MOVL    ROW, R2                                     ;
                              51            59  D0 0022B 43$:     MOVL    R9, R1                                      ;
                              50        24  AE  D0 0022F          MOVL    TEMP_STORE, R0                              ;
                              00000000G     00  16 00232          JSB     BAS$STO_FA_L_R8                             ;
                                            3E  11 00238          BRB     50$                                         ; 2820
                              06            6E  E9 0023A 44$:     BLBC    FLAGS, 45$                                  ; 2828
                              52        04  AE  D0 0023D          MOVL    COLUMN, R2                                  ; 2830
                                            08  11 00241          BRB     46$                                         ;
                              53        04  AE  D0 00243 45$:     MOVL    COLUMN, R3                                  ; 2832
                              52        08  AE  D0 00247          MOVL    ROW, R2                                     ;
                              51            59  D0 0024B 46$:     MOVL    R9, R1                                      ;
                              50        24  AE  D0 0024E          MOVL    TEMP_STORE, R0                              ;
                              00000000G     00  16 00252          JSB     BAS$STO_FA_F_R8                             ;
                                            6B  11 00258          BRB     55$                                         ; 2828
                              06            6E  E9 0025A 47$:     BLBC    FLAGS, 48$                                  ; 2836
                              53        04  AE  D0 0025D          MOVL    COLUMN, R3                                  ; 2838
                                            08  11 00261          BRB     49$                                         ;
                              54        04  AE  D0 00263 48$:     MOVL    COLUMN, R4                                  ; 2841
                              53        08  AE  D0 00267          MOVL    ROW, R3                                     ;
                              52            59  D0 0026B 49$:     MOVL    R9, R2                                      ;
                              50        24  AE  7D 0026E          MOVQ    TEMP_STORE, R0                              ;
                              00000000G     00  16 00272          JSB     BAS$STO_FA_D_R8                             ;
                                            6C  11 00278 50$:     BRB     59$                                         ; 2836
                              4A            6E  E9 0027A 51$:     BLBC    FLAGS, 56$                                  ; 2845
                                        04  AE  DD 0027D          PUSHL   COLUMN                                      ; 2847
                                            59  DD 00280          PUSHL   R9                                          ;
                                        2C  AE  9F 00282          PUSHAB  TEMP_STORE                                  ;
                                            58  11 00285          BRB     58$                                         ;
                16            06        02  AA  8F 00287 52$:     CASEB   2(ELEM_DESCRIP), #6, #22                    ; 2854
      002E      0048          0048          0048       0028C 53$:     .WORD   57$-53$,-                               ;
      002E      002E          0048          0048       00294                  57$-53$,-                               ;
      002E      002E          002E          FFEE       0029C                  57$-53$,-                               ;
      0048      002E          002E          002E       002A4                  54$-53$,-                               ;
      002E      002E          002E          002E       002AC                  57$-53$,-                               ;
                0048          0048          002E       002B4                  57$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              51$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              57$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              54$-53$,-                               ;
                                                                              57$-53$,-                               ;
                                                                              57$-53$                                 ;
                              7E       00G  8F   9A 002BA 54$:    MOVZBL  #BAS$K_DATTYPERR, -(SP)                     ; 2876
                   00000000G  00            01   FB 002BE         CALLS   #1, BAS$$STOP                               ;
```

```
                              77  11 002C5 55$:     BRB     68$                                     ;
                      04      AE  DD 002C7 56$:     PUSHL   COLUMN                                  ; 2872
                      0C      AE  DD 002CA          PUSHL   ROW                                     ;
                              59  DD 002CD          PUSHL   R9                                      ;
                      30      AE  9F 002CF          PUSHAB  TEMP_STORE                              ;
                              1F  11 002D2          BRB     61$                                     ;
              11              6E  E9 002D4 57$:     BLBC    FLAGS, 60$                              ; 2884
                      04      AE  DD 002D7          PUSHL   COLUMN                                  ; 2886
                              59  DD 002DA          PUSHL   R9                                      ;
                      24      AE  9F 002DC          PUSHAB  NUM_DESCRIP                             ;
     00000000G  00            03  FB 002DF 58$:     CALLS   #3, BAS$STORE_BFA                       ;
                              56  11 002E6 59$:     BRB     68$                                     ;
                      04      AE  DD 002E8 60$:     PUSHL   COLUMN                                  ; 2888
                      0C      AE  DD 002EB          PUSHL   ROW                                     ;
                              59  DD 002EE          PUSHL   R9                                      ;
                      28      AE  9F 002F0          PUSHAB  NUM_DESCRIP                             ;
     00000000G  00            04  FB 002F3 61$:     CALLS   #4, BAS$STORE_BFA                       ;
                              42  11 002FA          BRB     68$                                     ; 2884
              06              6E  E9 002FC 62$:     BLBC    FLAGS, 63$                              ; 2892
              53      04      AE  D0 002FF          MOVL    COLUMN, R3                              ; 2894
                              08  11 00303          BRB     64$                                     ;
              54      04      AE  D0 00305 63$:     MOVL    COLUMN, R4                              ; 2896
              53      08      AE  D0 00309          MOVL    ROW, R3                                 ;
              52              59  D0 0030D 64$:     MOVL    R9, R2                                  ;
              50      24      AE  7D 00310          MOVQ    TEMP_STORE, R0                          ;
                00000000G     00  16 00314          JSB     BAS$STO_FA_G_R8                         ;
                              22  11 0031A          BRB     68$                                     ; 2892
              06              6E  E9 0031C 65$:     BLBC    FLAGS, 66$                              ; 2900
              55      04      AE  D0 0031F          MOVL    COLUMN, R5                              ; 2902
                              08  11 00323          BRB     67$                                     ;
              56      04      AE  D0 00325 66$:     MOVL    COLUMN, R6                              ; 2905
              55      08      AE  D0 00329          MOVL    ROW, R5                                 ;
              54              59  D0 0032D 67$:     MOVL    R9, R4                                  ;
              52      2C      AE  7D 00330          MOVQ    TEMP_STORE+8, R2                        ;
              50      24      AE  7D 00334          MOVQ    TEMP_STORE, R0                          ;
                00000000G     00  16 00338          JSB     BAS$STO_FA_H_R8                         ;
                      18      AE  D6 0033E 68$:     INCL    NUM_ELEMS_DONE                          ; 2912
                      04      AE  D6 00341          INCL    COLUMN                                  ; 2913
                            FDE3  31 00344          BRW     20$                                     ; 2757
              07              6E  E9 00347 69$:     BLBC    FLAGS, 70$                              ; 2916
52        04  AE              01  C3 0034A          SUBL3   #1, COLUMN, R2                          ;
                              04  11 0034F          BRB     71$                                     ;
              52      08      AE  D0 00351 70$:     MOVL    ROW, R2                                 ;
     00000000'  EF            52  D0 00355 71$:     MOVL    R2, NUM                                 ;
              04              6E  E9 0035C          BLBC    FLAGS, 72$                              ; 2917
                              52  D4 0035F          CLRL    R2                                      ;
                              05  11 00361          BRB     73$                                     ;
52        04  AE              01  C3 00363 72$:     SUBL3   #1, COLUMN, R2                          ;
     00000000'  EF            52  D0 00368 73$:     MOVL    R2, NUM2                                ;
              0E      0C      BE  91 0036F          CMPB    @12(SP), #14                            ; 2922
                              0C  13 00373          BEQL    74$                                     ;
              18      0C      BE  91 00375          CMPB    @12(SP), #24                            ; 2923
                              10  12 00379          BNEQ    75$                                     ;
              0E      02      AA  91 0037B          CMPB    2(ELEM_DESCRIP), #14                    ; 2924
                              0A  12 0037F          BNEQ    75$                                     ;
                      24      AE  9F 00381 74$:     PUSHAB  TEMP_STORE                              ; 2926
     00000000G  00            01  FB 00384          CALLS   #1, STR$FREE1_DX                        ;
```

```
                                             04 00388 75$:     RET                                                 ; 2929

; Routine Size:  908 bytes,    Routine Base:  _BAS$CODE + 0A8A


; 2015          2930  1
; 2016          2931  1 END                                                !End of module - BAS$MAT_IO
; 2017          2932  1
; 2018          2933  0 ELUDOM




;                                PSECT SUMMARY
;
;       Name                        Bytes                        Attributes
;
;    _BAS$DATA                          8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;    _BAS$CODE                       3606  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        18         0         581        00:01.2




;                                COMMAND QUALIFIERS

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

; Size:          3606 code + 8 data bytes
; Run Time:         01:08.1
; Elapsed Time:     02:23.2
; Lines/CPU Min:     2583
; Lexemes/CPU-Min: 21299
; Memory Used:  367 pages
; Compilation Complete
```

0026 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASMATMUL
LIS

BASMATIO
LIS

BASMATRED
LIS

BASMATNUL
LIS

BASMATINV
LIS