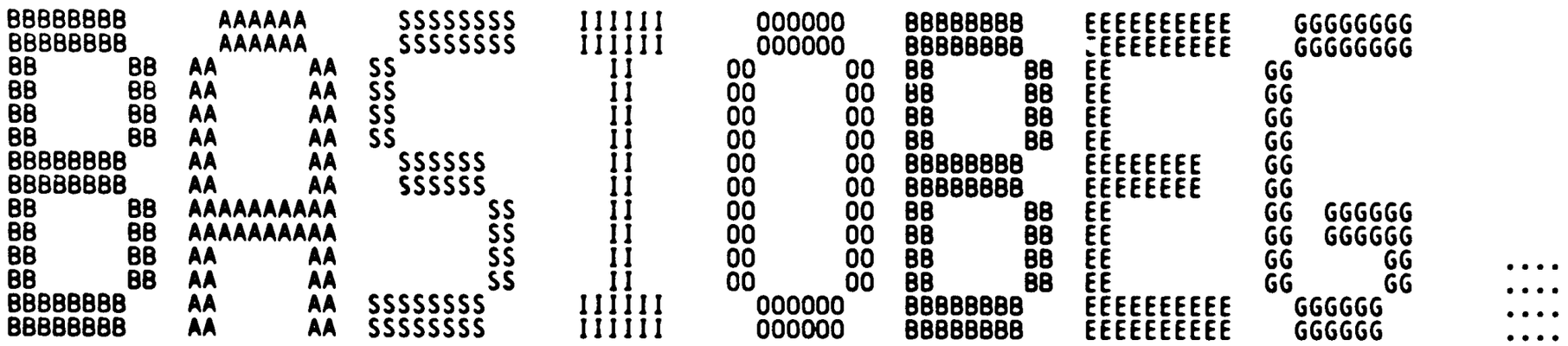
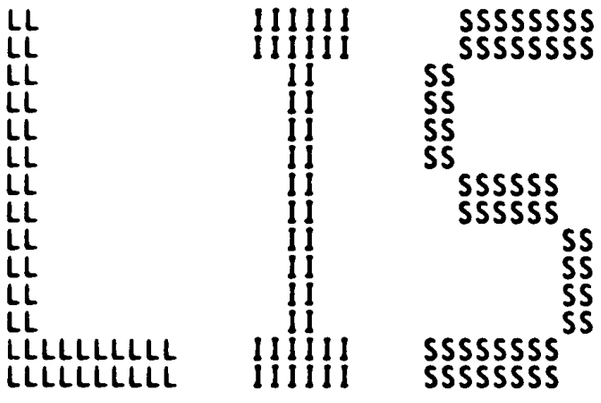
M 1

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


BBBBBBBB AAAAAA SSSSSSSS IIIIII OOOOOO BBBBBBBB EEEEEEEEEE GGGGGGGG
BBBBBBBB AAAAAA SSSSSSSS IIIIII OOOOOO BBBBBBBB EEEEEEEEEE GGGGGGGG
BB BB AA AA SS II OO OO BB BB EE GG
BB BB AA AA SS II OO OO BB BB EE GG
BB BB AA AA SS II OO OO BB BB EE GG
BB BB AA AA SS II OO OO BB BB EE GG
BBBBBBBB AA AA SSSSSS II OO OO BBBBBBBB EEEEEEEE GG
BBBBBBBB AA AA SSSSSS II OO OO BBBBBBBB EEEEEEEE GG
BB BB AAAAAAAAAA SS II OO OO BB BB EE GG GGGGGG
BB BB AAAAAAAAAA SS II OO OO BB BB EE GG GGGGGG
BB BB AA AA SS II OO OO BB BB EE GG GG
BB BB AA AA SS II OO OO BB BB EE GG GG
BBBBBBBB AA AA SSSSSSSS IIIIII OOOOOO BBBBBBBB EEEEEEEEEE GGGGGG
BBBBBBBB AA AA SSSSSSSS IIIIII OOOOOO BBBBBBBB EEEEEEEEEE GGGGGG


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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1          0001  0 MODULE BAS$IO_BEG (                    ! BASIC READ/WRITE statement initialization user calls
;     2          0002  0                 IDENT = '1-057'        ! File: BASIOBEG.B32 Edit:MDL1057
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************
;     7          0007  1 !*                                                                        *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                   *
;    11          0011  1 !*                                                                        *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                           *
;    18          0018  1 !*                                                                        *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                           *
;    22          0022  1 !*                                                                        *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !**************************************************************************
;    28          0028  1
;    29          0029  1 !++
;    30          0030  1 ! FACILITY: BASIC + 2 Support LIbrary - User callable
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !
;    34          0034  1 !       This module (plus BAS$IO_ELM and BAS$IO_END modules)
;    35          0035  1 !       implements BASIC READ/WRITE (INPUT
;    36          0036  1 !       and PRINT) statements at the user program interface level
;    37          0037  1 !       of abstraction. (UPI = first level). To implement
;    38          0038  1 !       a single I/O statement a user program calls one of the
;    39          0039  1 !       initialization routines in this module followed by zero
;    40          0040  1 !       or more calls to be BAS$IO_ELM module and terminated by a
;    41          0041  1 !       call to the BAS$IO_END module. This module also defines the
;    42          0042  1 !       vectors for the 2nd and 3rd levels of abstraction (used
;    43          0043  1 !       data formatters (UDF) and record processors (REC).
;    44          0044  1 !
;    45          0045  1 ! ENVIRONMENT: User access mode; mixture of AST level or not.
;    46          0046  1 !
;    47          0047  1 ! AUTHOR: Donald G. Petersen, CREATION DATE: 14-Mar-78: Version 01
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 ! 0-37  - Don't set the record number into LOG_RECNO until after OPEN,
;    52          0052  1 !         since OPEN sets it to 1.  JMT 27-Jan-78
;    53          0053  1 !       Donald G. Petersen, 14-Mar-78 : VERSION 1-01
;    54          0054  1 ! 01    - original
;    55          0055  1 ! 02    - debug.  DGP 06-Jun-78 1430
;    56          0056  1 ! 03    - change dispatch tables to OWN storage.  DGP 06-Jun-78 1530
;    57          0057  1 ! 1-04  - Make PRINT and INPUT logical unit Nos. -1 & -2. DGP 07-Jun-78
```

```
;    58          0058  1 ! 1-05  - change to JSB linkages.  DGP 14-Nov-78
;    59          0059  1 ! 1-06  - Change names of require files.  DGP 28-Nov-78
;    60          0060  1 ! 1-007 - Add device names to REQUIRE files and update copyright
;    61          0061  1 !         notice.  JBS 29-NOV-78
;    62          0062  1 ! 1-008 - Change call to OPEN_DEFLT.  Remove FORMATTED arg.  DGP 04-Dec-78
;    63          0063  1 ! 1-009 - Change require of BASOPN to FOROPN.  DGP 05-Dec-78
;    64          0064  1 ! 1-010 - Change REQUIRE file names from FOR... to OTS...  JBS 06-DEC-78
;    65          0065  1 ! 1-011 - Change the statement types passed by the UPI to conform with new
;    66          0066  1 !         names.  DGP 07-Dec-78
;    67          0067  1 ! 1-012 - Change statement types to ISB prefix.  DGP 08-Dec-78
;    68          0068  1 ! 1-013 - Add READ.  DGP 12-Dec-78
;    69          0069  1 ! 1-014 - Modify whole module to eliminate primitive BLISS argument passing
;    70          0070  1 !         techniques.  DGP 13-Dec-78
;    71          0071  1 ! 1-015 - Add some more stuff for READ.  DGP 19-Dec-78
;    72          0072  1 ! 1-016 - Change call to CB_PUSH to pass ILUN_MIN instead of DLUN_MIN. DGP
;    73          0073  1 !         21-Dec-78
;    74          0074  1 ! 1-017 - Change call to FOR$$CB_PUSH to call BAS$$CB_PUSH.  JBS 29-DEC-78
;    75          0075  1 ! 1-018 - Change ISB$A_BUF_PTR, BUF_BEG, BUF_END to LUB.  DGP 05-Jan-79
;    76          0076  1 ! 1-019 - Remove REQUIRE of BASPAR--not needed.  JBS 10-JAN-1979
;    77          0077  1 ! 1-020 - Bug fix to signal I/O channel not open logic. DGP 10-Jan-79
;    78          0078  1 ! 1-021 - Add code to BAS$INPUT to handle prompting and 'dirty' Print buffers
;    79          0079  1 !         in recursive I/O situations.  DGP 13-Jan-79
;    80          0080  1 ! 1-022 - Fix bug in buddy pointer handling.  DGP 19-Jan-79
;    81          0081  1 ! 1-023 - Reset Print buffer pointer to BUF_BEG if contents are transferred
;    82          0082  1 !         to the Prompt buffer.  DGP 24-Jan-79
;    83          0083  1 ! 1-024 - Clear ISB$W_FMT_LEN, so IO_END will not try to deallocate
;    84          0084  1 !         a supposed object-time format.  JBS 12-MAR-1979
;    85          0085  1 ! 1-025 - Fix a bug in BUDDY_PTR assignment on non-zero channels.  DGP 05-Apr-79
;    86          0086  1 ! 1-026 - A check is made for a terminal device before doing Prompt business.
;    87          0087  1 !         DGP 06-Apr-79
;    88          0088  1 ! 1-027 - Use BAS$$OPEN_ZERO to open channel 0 instead of calling
;    89          0089  1 !         BAS$$OPEN_DEFLT twice.  JBS 17-APR-1979
;    90          0090  1 ! 1-028 - Store address of start of I/O list in ISB$A_RESTARTPC so
;    91          0091  1 !         that I/O statements can be restarted.  JBS 07-MAY-1979
;    92          0092  1 ! 1-029 - Add Basic PRINT USING.  DGP 14-May-79
;    93          0093  1 ! 1-030 - Make a few more changes for PRINT USING.  DGP 21-May-79
;    94          0094  1 ! 1-031 - Correct a typo in PRINT USING that made all I/O
;    95          0095  1 !         statements fail.  JBS 22-MAY-1979
;    96          0096  1 ! 1-032 - Initialize ISB$W_LEN_REM.  DGP 22-May-79
;    97          0097  1 ! 1-033 - Add MAT PRINT.  DGP 15-Jun-79
;    98          0098  1 ! 1-034 - Use language-specific dispatch tables.  JBS 26-JUN-1979
;    99          0099  1 ! 1-035 - Check the language byte in the LUB to verify that
;   100          0100  1 !         the file was opened by BASIC.  JBS 30-JUN-1979
;   101          0101  1 ! 1-036 - Use ISB symbols for dispatching.  JBS 12-JUL-1979
;   102          0102  1 ! 1-037 - Add BAS$MAT_READ, BAS$MAT_LINPUT.  DGP 13-Jul-79
;   103          0103  1 ! 1-038 - Remove edit 035: the language check is now being done
;   104          0104  1 !         in BAS$$CB.  JBS 14-JUL-1979
;   105          0105  1 ! 1-039 - Store caller's FP in the ISB, so we can call BAS$$CB_POP
;   106          0106  1 !         after an I/O error (from BAS$$UNWIND), and do not depend
;   107          0107  1 !         on inheriting R11.  JBS 24-JUL-1979
;   108          0108  1 ! 1-040 - Change call to BAS$$OPEN_ZERO.  JBS 26-JUL-1979
;   109          0109  1 ! 1-041 - Remove the reference to BAS$$SIGDIS_ERR.  JBS 01-AUG-1979
;   110          0110  1 ! 1-042 - Dirty buffer check now checks for Print Using as well as Print
;   111          0111  1 !         DGP 01-Aug-79
;   112          0112  1 ! 1-043 - These I/O statements are allowed only on terminal format and
;   113          0113  1 !         sequential files.  JBS 01-AUG-1979
;   114          0114  1 ! 1-044 - Improve edit 043 by testing LUB$V_NOTSEQORG.  JBS 08-JUL-1979
```

```
;  115          0115  1 ! 1-045 - MAT_PRINT now passes unit number by value.  DGP 06-Sep-79
;  116          0116  1 ! 1-046 - Check for VIRTUAL usage.  Set_BLOCK usage.  PRINT and others like
;  117          0117  1 !         it check for READONLY.  DGP 13-Sep-79
;  118          0118  1 ! 1-047 - Use the proper statement types for MAT READ and MAT LINPUT.
;  119          0119  1 !         DGP 10-Oct-79
;  120          0120  1 ! 1-048 - Stuff the scale factor into the ISB.  DGP  15-Nov-79
;  121          0121  1 ! 1-049 - Use new name ISB$B_SCA_FAC_D.  DGP 26-Nov-79
;  122          0122  1 ! 1-050 - Both Input and Print only store the -6 to 0 scale factor in the
;  123          0123  1 !         ISB.  DGP 28-Nov-79
;  124          0124  1 ! 1-051 - Clear the print format character flag if moving it into the prompt
;  125          0125  1 !         buffer.  DGP 10-Dec-79
;  126          0126  1 ! 1-052 - Clear RAB$B_PSZ and LUB$L_PRINT_POS in BAS$$IO_BEG to cure the problem of
;  127          0127  1 !         concatenating to the prompt buffer the same prompt in case of
;  128          0128  1 !         ^C trapping and resume to that input statement.
;  129          0129  1 !         At the same time we have taken out the
;  130          0130  1 !         statement in PUSH_ACTIVE routine that clears this byte to keep
;  131          0131  1 !         the locality consistent. FM 4-SEP-1980
;  132          0132  1 ! 1-053 - Check for negative channel numbers passed from compiled code in every
;  133          0133  1 !         routine except BAS$MAT_READ and BAS$_READ, if found signal ILLIO_CHA.
;  134          0134  1 !         FM 9-SEP-1980
;  135          0135  1 ! 1-054 - Cursor position is not updated properly on INPUT if the previous
;  136          0136  1 !         PRINT terminator was a semicolon or comma.  PLL 7-May-81
;  137          0137  1 ! 1-055 - Add support for ANSI INPUT.  PLL 22-Jul-1982
;  138          0138  1 ! 1-056 - Add support for ANSI PRINT.  MDL 1-Apr-1983
;  139          0139  1 ! 1-057 - BAS$$IO_BEG, reset BUF_PTR to BUF_BEG only if this is a terminal
;  140          0140  1 !         device.  This allows the REC level routine to write the contents
;  141          0141  1 !         of the Print buffer to the batch log, thus simulating prompts.
;  142          0142  1 !         MDL 27-Jul-1983
;  143          0143  1 !--
;  144          0144  1
;  145          0145  1 !<BLF/PAGE>
```

```
;  147          0146  1 !
;  148          0147  1 ! SWITCHES:
;  149          0148  1 !
;  150          0149  1
;  151          0150  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;  152          0151  1
;  153          0152  1 !
;  154          0153  1 ! LINKAGES
;  155          0154  1 !
;  156          0155  1
;  157          0156  1 REQUIRE 'RTLIN:OTSLNK';                     ! Define all linkages
;  158          0585  1 REQUIRE 'RTLIN:BASLNK';                ! Some BASIC specific linkages
;  159          0662  1
;  160          0663  1 !
;  161          0664  1 ! TABLE OF CONTENTS:
;  162          0665  1 !
;  163          0666  1
;  164          0667  1 FORWARD ROUTINE
;  165          0668  1     BAS$MAT_LINPUT : NOVALUE,                   ! MAT LINPUT
;  166          0669  1     BAS$MAT_READ : NOVALUE,                     ! MAT READ
;  167          0670  1     BAS$MAT_INPUT : NOVALUE,                    ! MAT INPUT
;  168          0671  1     BAS$MAT_PRINT : NOVALUE,                    ! MAT PRINT
;  169          0672  1     BAS$PRINT_USING : NOVALUE,                  ! PRINT USING
;  170          0673  1     BAS$READ : NOVALUE,                         ! READ data
;  171          0674  1     BAS$LINPUT : NOVALUE,                       ! READ (LINPUT)
;  172          0675  1     BAS$INPUT_LINE : NOVALUE,                   ! READ (INPUT LINE)
;  173          0676  1     BAS$INPUT : NOVALUE,                        ! READ sequential list-directed
;  174          0677  1     BAS$ANSI_INPUT : NOVALUE,                   ! same as INPUT, but ANSI standard
;  175          0678  1     BAS$PRINT : NOVALUE,                        ! WRITE sequential list-directed
;  176          0679  1     BAS$ANSI_PRINT : NOVALUE,                   ! same as PRINT, but ANSI standard
;  177          0680  1     BAS$$IO_BEG : NOVALUE;                      ! Common routine for all I/O statements
;  178          0681  1
;  179          0682  1 !
;  180          0683  1 ! INCLUDE FILES:
;  181          0684  1 !
;  182          0685  1
;  183          0686  1 REQUIRE 'RTLML:OTSLUB';                     ! logical unit block (LUB) offsets
;  184          0826  1
;  185          0827  1 REQUIRE 'RTLML:OTSISB';                     ! I/O statement block (ISB) offsets
;  186          0995  1
;  187          0996  1 REQUIRE 'RTLIN:OTSMAC';                     ! Macros
;  188          1190  1
;  189          1191  1 REQUIRE 'RTLIN:RTLPSECT';                   ! Define DECLARE_PSECTS macro
;  190          1286  1
;  191          1287  1 REQUIRE 'RTLIN:BASFRAME';                   ! BASIC frame structure
;  192          1490  1
;  193          1491  1 LIBRARY 'RTLSTARLE';                        ! STARLET macros and symbols
;  194          1492  1
;  195          1493  1 !
;  196          1494  1 ! MACROS:
;  197          1495  1 !
;  198          1496  1 !       NONE
;  199          1497  1 !
;  200          1498  1 ! EQUATED SYMBOLS:
;  201          1499  1 !
;  202          1500  1 !       NONE
;  203          1501  1 !
```

```
;  204          1502  1 ! PSECT DECLARATIONS:
;  205          1503  1 !
;  206          1504  1 DECLARE_PSECTS (BAS);
;  207          1505  1 !
;  208          1506  1 ! OWN STORAGE:
;  209          1507  1 !A
;  210          1508  1 !       NONE
;  211          1509  1 !
;  212          1510  1 !
;  213          1511  1 ! EXTERNAL REFERENCES:
;  214          1512  1 !
;  215          1513  1
;  216          1514  1 EXTERNAL LITERAL
;  217          1515  1     BAS$K_ILLILLACC : UNSIGNED (8),          ! Illogical or illegal access
;  218          1516  1     BAS$K_IO_CHANOT : UNSIGNED (8),          ! I/O channel not open
;  219          1517  1     BAS$K_TERFORFIL : UNSIGNED (8),          ! Terminal format file required
;  220          1518  1     BAS$K_ILLIO_CHA : UNSIGNED (8);          ! Illegal I/O channel
;  221          1519  1
;  222          1520  1 EXTERNAL ROUTINE
;  223          1521  1     BAS$$SCALE_L_R1 : BAS$SCALE_JSB,         ! Fetch the scale factor
;  224          1522  1     BAS$$SCALE_RT : BAS$SCALE_JSB,           ! Fetch a scale factor
;  225          1523  1     BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,      ! Load register CCB
;  226          1524  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,        ! Done with register CCB
;  227          1525  1     BAS$$STOP_IO : NOVALUE,                  ! Signal a fatal BASIC I/O error
;  228          1526  1     BAS$$OPEN_ZERO : NOVALUE,                ! Open channel zero.
;  229          1527  1     BAS$$STOP : NOVALUE;                     ! Signal a fatal error.
;  230          1528  1
;  231          1529  1 !+
;  232          1530  1 ! User data formatter level of abstraction (UDF is 2nd level)
;  233          1531  1 !-
;  234          1532  1
;  235          1533  1 EXTERNAL
;  236          1534  1     BAS$$AA_UDF_PRO : VECTOR;                ! Dispatch table for UDF level
;  237          1535  1
;  238          1536  1 !+
;  239          1537  1 ! This value, if in 0(FP), marks the frame as a BASIC frame.
;  240          1538  1 !-
;  241          1539  1
;  242          1540  1 EXTERNAL
;  243          1541  1     BAS$HANDLER;
;  244          1542  1
;  245          1543  1 LITERAL
;  246          1544  1     K_PRE_COM_FLAG = 1,                      ! if set pre-compiled PRINT USING string
;  247          1545  1     K_ANSI_FLAG = 2;                         ! if set ANSI error checking
;  248          1546  1
```

```
;  250          1547  1 GLOBAL ROUTINE BAS$LINPUT (                    ! READ sequential list-directed
;  251          1548  1         UNIT                                   ! logical unit
;  252          1549  1     ) : NOVALUE =
;  253          1550  1
;  254          1551  1 !++
;  255          1552  1 ! ABSTRACT:
;  256          1553  1 !
;  257          1554  1 !       Initialize the BASIC I/O system to perform
;  258          1555  1 !       a READ sequential list-directed.
;  259          1556  1 !
;  260          1557  1 ! FORMAL PARAMETERS:
;  261          1558  1 !
;  262          1559  1 !       UNIT.rl.v       logical unit number
;  263          1560  1 !
;  264          1561  1 ! IMPLICIT INPUTS:
;  265          1562  1 !
;  266          1563  1 !       NONE
;  267          1564  1 !
;  268          1565  1 ! IMPLICIT OUTPUTS:
;  269          1566  1 !
;  270          1567  1 !       NONE
;  271          1568  1 !
;  272          1569  1 ! ROUTINE VALUE:
;  273          1570  1 !
;  274          1571  1 !       NONE
;  275          1572  1 !
;  276          1573  1 ! SIDE EFFECTS:
;  277          1574  1 !
;  278          1575  1 !       LUB/ISB/RAB control blocks allocated and file
;  279          1576  1 !       opened if not already. SIGNAL_STOPs many errors.
;  280          1577  1 !--
;  281          1578  1
;  282          1579  2     BEGIN
;  283          1580  2
;  284          1581  2     BUILTIN
;  285          1582  2         FP;
;  286          1583  2
;  287          1584  2     LOCAL
;  288          1585  2         FMP : REF BLOCK [, BYTE];
;  289          1586  2
;  290          1587  2     FMP = .FP;
;  291          1588  2 !+
;  292          1589  2 ! If channel number is less than zero then it is an illegal channel.
;  293          1590  2 !-
;  294          1591  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  295          1592  2
;  296          1593  2 !+
;  297          1594  2 ! Perform default OPEN if UNIT not already opened.
;  298          1595  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  299          1596  2 ! any errors.  Store I/O statement type code. Call
;  300          1597  2 ! UDF level initialization.
;  301          1598  2 !-
;  302          1599  2     BAS$$IO_BEG (ISB$K_ST_TY_LIN, (IF (.UNIT EQL 0) THEN LUB$K_LUN_INPU ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  303          1600  2         .FMP [SF$L_SAVE_P[]);
;  304          1601  1     END;
```

```
                                        .TITLE  BAS$IO_BEG
                                        .IDENT  \1-057\

                                        .EXTRN  BAS$K_ILLILLACC
                                        .EXTRN  BAS$K_IO_CHANOT
                                        .EXTRN  BAS$K_TERFORFIL
                                        .EXTRN  BAS$K_ILLIO_CHA
                                        .EXTRN  BAS$$SCALE_C_R1
                                        .EXTRN  BAS$$SCALE_RT, BAS$$CB_PUSH
                                        .EXTRN  BAS$$CB_POP, BAS$$STOP_IO
                                        .EXTRN  BAS$$OPEN_ZERO, BAS$$STOP
                                        .EXTRN  BAS$$AA_UDF_PRO
                                        .EXTRN  BAS$HANDLER

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

                          000C 00000    .ENTRY  BAS$LINPUT, Save R2,R3                  ; 1547
                    52    5D   D0 00002 MOVL    FP, FMP                               ; 1587
                    53 04 AC   D0 00005 MOVL    UNIT, R3                              ; 1591
                       08 18   00009    BGEQ    1$                                    ;
                 7E    00G 8F   9A 0000B MOVZBL  #BAS$K_ILLIO_CHA, -(SP)               ;
00000000G        00       01   FB 0000F CALLS   #1, BAS$$STOP                         ;
                 7E    0C A2   7D 00016 1$:    MOVQ    12(FMP), -(SP)                ; 1599
                       53   D5 0001A    TSTL    R3                                    ;
                       05   12 0001C    BNEQ    2$                                    ;
                 7E         07   CE 0001E MNEGL   #7, -(SP)                           ;
                       02   11 00021    BRB     3$                                    ;
                       53   DD 00023 2$: PUSHL   R3                                   ;
                       1C   DD 00025 3$: PUSHL   #28                                 ;
                0000V  CF         04   FB 00027 CALLS   #4, BAS$$IO_BEG                ;
                                  04 0002C    RET                                     ; 1601

; Routine Size:  45 bytes,    Routine Base:  _BAS$CODE + 0000

;  305          1602  1
```

```
;  307          1603  1 GLOBAL ROUTINE BAS$MAT_LINPUT (                  ! READ sequential list-directed
;  308          1604  1         UNIT                                     ! logical unit
;  309          1605  1     ) : NOVALUE =
;  310          1606  1
;  311          1607  1 !++
;  312          1608  1 ! ABSTRACT:
;  313          1609  1 !
;  314          1610  1 !       Initialize the BASIC I/O system to perform
;  315          1611  1 !       a READ sequential list-directed.
;  316          1612  1 !
;  317          1613  1 ! FORMAL PARAMETERS:
;  318          1614  1 !
;  319          1615  1 !       UNIT.rl.v       logical unit number
;  320          1616  1 !
;  321          1617  1 ! IMPLICIT INPUTS:
;  322          1618  1 !
;  323          1619  1 !       NONE
;  324          1620  1 !
;  325          1621  1 ! IMPLICIT OUTPUTS:
;  326          1622  1 !
;  327          1623  1 !       NONE
;  328          1624  1 !
;  329          1625  1 ! ROUTINE VALUE:
;  330          1626  1 !
;  331          1627  1 !       NONE
;  332          1628  1 !
;  333          1629  1 ! SIDE EFFECTS:
;  334          1630  1 !
;  335          1631  1 !       LUB/ISB/RAB control blocks allocated and file
;  336          1632  1 !       opened if not already. SIGNAL_STOPs many errors.
;  337          1633  1 !--
;  338          1634  1
;  339          1635  2     BEGIN
;  340          1636  2
;  341          1637  2     BUILTIN
;  342          1638  2         FP;
;  343          1639  2
;  344          1640  2     LOCAL
;  345          1641  2         FMP : REF BLOCK [, BYTE];
;  346          1642  2
;  347          1643  2     FMP = .FP;
;  348          1644  2 !+
;  349          1645  2 ! If channel number is less than zero then it is an illegal channel.
;  350          1646  2 !-
;  351          1647  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  352          1648  2
;  353          1649  2 !+
;  354          1650  2 ! Perform default OPEN if UNIT not already  pened.
;  355          1651  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  356          1652  2 ! any errors.  Store I/O statement type code. Call
;  357          1653  2 ! UDF level initialization.
;  358          1654  2 !-
;  359          1655  2     BAS$$IO_BEG (ISB$K_ST_TY_MLI, (IF (.UNIT EQL 0) THEN LUB$K_LUN_INPU ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  360          1656  2         .FMP [SF$L_SAVE_PC]);
;  361          1657  1     END;
```

```
                                    000C 00000          .ENTRY  BAS$MAT_LINPUT, Save R2,R3                      ; 1603
                            52      5D   D0 00002          MOVL    FP, FMP                                         ; 1643
                            53  04  AC   D0 00005          MOVL    UNIT, R3                                        ; 1647
                                    0B   18 00009          BGEQ    1$                                              ;
                            7E  00G 8F   9A 0000B          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                         ;
                00000000G   00      01   FB 0000F          CALLS   #1, BAS$$STOP                                   ;
                            7E  0C  A2   7D 00016 1$:      MOVQ    12(FMP), -(SP)                                  ; 1655
                                    53   D5 0001A          TSTL    R3                                              ;
                                    05   12 0001C          BNEQ    2$                                              ;
                            7E      07   CE 0001E          MNEGL   #7, -(SP)                                       ;
                                    02   11 00021          BRB     3$                                              ;
                                    53   DD 00023 2$:      PUSHL   R3                                              ;
                                    32   DD 00025 3$:      PUSHL   #50                                             ;
                      0000V CF              04   FB 00027  CALLS   #4, BAS$$IO_BEG                                 ;
                                            04   0002C    RET                                                     ; 1657

; Routine Size:  45 bytes,    Routine Base:  _BAS$CODE + 002D

;  362          1658  1
```

```
;  364          1659  1 GLOBAL ROUTINE BAS$READ                              ! READ
;  365          1660  1     : NOVALUE =
;  366          1661  1
;  367          1662  1 !++
;  368          1663  1 ! ABSTRACT:
;  369          1664  1 !
;  370          1665  1 !       Initialize the BASIC I/O system to perform
;  371          1666  1 !       a READ of a compile time initialized DATA area.
;  372          1667  1 !
;  373          1668  1 ! FORMAL PAREMETERS:
;  374          1669  1 !
;  375          1670  1 !       NONE
;  376          1671  1 !
;  377          1672  1 ! IMPLICIT INPUTS:
;  378          1673  1 !
;  379          1674  1 !       R11.rl.v                pointer to most recent Basic major frame
;  380          1675  1 !
;  381          1676  1 ! IMPLICIT OUTPUTS:
;  382          1677  1 !
;  383          1678  1 !       ISB$L_MAJ_F_PTR         Basic major frame pointer
;  384          1679  1 !
;  385          1680  1 ! ROUTINE VALUE:
;  386          1681  1 !
;  387          1682  1 !       NONE
;  388          1683  1 !
;  389          1684  1 ! SIDE EFFECTS:
;  390          1685  1 !
;  391          1686  1 !       LUB/ISB/RAB control blocks allocated and file
;  392          1687  1 !       opened if not already.
;  393          1688  1 !--
;  394          1689  1
;  395          1690  2     BEGIN
;  396          1691  2
;  397          1692  2     BUILTIN
;  398          1693  2         FP;
;  399          1694  2
;  400          1695  2     LOCAL
;  401          1696  2         FMP : REF BLOCK [, BYTE];
;  402          1697  2
;  403          1698  2     FMP = .FP;
;  404          1699  2 !+
;  405          1700  2 ! Perform default OPEN if UNIT not already opened.
;  406          1701  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  407          1702  2 ! any errors.  Store I/O statement type code. Call
;  408          1703  2 ! UDF level initialization.
;  409          1704  2 !-
;  410          1705  2     BAS$$IO_BEG (ISB$K_ST_TY_REA, LUB$K_LUN_BREAD, .FMP [SF$L_SAVE_FP], .FMP [SF$L_SAVE_PC]);
;  411          1706  1     END;


                                0000 00000          .ENTRY  BAS$READ, Save nothing                  ; 1659
                            5D  D0 00002            MOVL    FP, FMP                                 ; 1698
7E          0C              A0  7D 00005            MOVQ    12(FMP), -(SP)                          ; 1705
7E                          06  CE 00009            MNEGL   #6, -(SP)                               ;
```

```
                                  22  DD 0000C        PUSHL   #34                          ;
                0000V  CF         04  FB 0000E        CALLS   #4, BAS$$IO_BEG              ;
                                      04 00013        RET                                  ; 1706

; Routine Size:  20 bytes,    Routine Base:  _BAS$CODE + 005A

;  412          1707  1
```

```
;  414     1708  1 GLOBAL ROUTINE BAS$MAT_READ                          ! MAT READ
;  415     1709  1     : NOVALUE =
;  416     1710  1
;  417     1711  1 !++
;  418     1712  1 ! ABSTRACT:
;  419     1713  1 !
;  420     1714  1 !       Initialize the BASIC I/O system to perform
;  421     1715  1 !       a READ of a comp·le time initialized DATA area.
;  422     1716  1 !
;  423     1717  1 ! FORMAL PAREMETERS:
;  424     1718  1 !
;  425     1719  1 !       NONE
;  426     1720  1 !
;  427     1721  1 ! IMPLICIT INPUTS:
;  428     1722  1 !
;  429     1723  1 !       R11.rl.v                pointer to most recent Basic major frame
;  430     1724  1 !
;  431     1725  1 ! IMPLICIT OUTPUTS:
;  432     1726  1 !
;  433     1727  1 !       ISB$L_MAJ_F_PTR         Basic major frame pointer
;  434     1728  1 !
;  435     1729  1 ! ROUTINE VALUE:
;  436     1730  1 !
;  437     1731  1 !       NONE
;  438     1732  1 !
;  439     1733  1 ! SIDE EFFECTS:
;  440     1734  1 !
;  441     1735  1 !       LUB/ISB/RAB control blocks allocated and file
;  442     1736  1 !       opened if not already.
;  443     1737  1 !--
;  444     1738  1
;  445     1739  2     BEGIN
;  446     1740  2
;  447     1741  2     BUILTIN
;  448     1742  2         FP;
;  449     1743  2
;  450     1744  2     LOCAL
;  451     1745  2         FMP : REF BLOCK [, BYTE];
;  452     1746  2
;  453     1747  2     FMP = .FP;
;  454     1748  2 !+
;  455     1749  2 ! Perform default OPEN if UNIT not already opened.
;  456     1750  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  457     1751  2 ! any errors.  Store I/O statement type code. Call
;  458     1752  2 ! UDF level initialization.
;  459     1753  2 !-
;  460     1754  2     BAS$$IO_BEG (ISB$K_ST_TY_MRE, LUB$K_LUN_BREAD, .FMP [SF$L_SAVE_FP], .FMP [SF$L_SAVE_PC]);
;  461     1755  1     END;                                ! end of BAS$MAT_READ


                                0000 00000          .ENTRY  BAS$MAT_READ, Save nothing                  ; 1708
                          5D    D0 00002          MOVL    FP, FMP                                     ; 1747
50                        5D    D0 00002          MOVL    FP, FMP
7E              0C        A0    7D 00005          MOVQ    12(FMP), -(SP)                              ; 1754
7E                        06    CE 00009          MNEGL   #6, -(SP)                                   ;
```

```
                                 36  DD 0000C         PUSHL   #54                                          ;
                      0000V  CF  04  FB 0000E         CALLS   #4, BAS$$IO_BEG                              ;
                                 04 00013         RET                                                     ; 1755

; Routine Size:  20 bytes,    Routine Base:  _BAS$CODE + 006E

;  462          1756  1
```

```
;  464     1757  1 GLOBAL ROUTINE BAS$INPUT_LINE (                  ! READ sequential list-directed
;  465     1758  1         UNIT                                      ! logical unit
;  466     1759  1     ) : NOVALUE =
;  467     1760  1
;  468     1761  1 !++
;  469     1762  1 ! ABSTRACT:
;  470     1763  1 !
;  471     1764  1 !       Initialize the BASIC I/O system to perform
;  472     1765  1 !       a READ sequential list-directed.
;  473     1766  1 !
;  474     1767  1 ! FORMAL PAREMETERS:
;  475     1768  1 !
;  476     1769  1 !       UNIT.rl.v       logical unit number
;  477     1770  1 !
;  478     1771  1 ! IMPLICIT INPUTS:
;  479     1772  1 !
;  480     1773  1 !       NONE
;  481     1774  1 !
;  482     1775  1 ! IMPLICIT OUTPUTS:
;  483     1776  1 !
;  484     1777  1 !       NONE
;  485     1778  1 !
;  486     1779  1 ! ROUTINE VALUE:
;  487     1780  1 !
;  488     1781  1 !       NONE
;  489     1782  1 !
;  490     1783  1 ! SIDE EFFECTS:
;  491     1784  1 !
;  492     1785  1 !       LUB/ISB/RAB control blocks allocated and file
;  493     1786  1 !       opened if not already. SIGNAL_STOPs many errors.
;  494     1787  1 !--
;  495     1788  1
;  496     1789  2     BEGIN
;  497     1790  2
;  498     1791  2     BUILTIN
;  499     1792  2         FP;
;  500     1793  2
;  501     1794  2     LOCAL
;  502     1795  2         FMP : REF BLOCK [, BYTE];
;  503     1796  2
;  504     1797  2     FMP = .FP;
;  505     1798  2 !+
;  506     1799  2 ! If channel number is less than zero then it is an illegal channel.
;  507     1800  2 !-
;  508     1801  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  509     1802  2
;  510     1803  2 !+
;  511     1804  2 ! Perform default OPEN if UNIT not already opened.
;  512     1805  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  513     1806  2 ! any errors.  Store I/O statement type code. Call
;  514     1807  2 ! UDF level initialization.
;  515     1808  2 !-
;  516     1809  2     BAS$$IO_BEG (ISB$K_ST_TY_INL, (IF (.UNIT EQL 0) THEN LUB$K_LUN_INPU ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  517     1810  2         .FMP [SF$L_SAVE_PC]);
;  518     1811  1     END;
```

```
                                 000C 00000          .ENTRY  BAS$INPUT_LINE, Save R2,R3                      ; 1757
                          52     5D   D0 00002       MOVL    FP, FMP                                         ; 1797
                          53  04 AC   D0 00005       MOVL    UNIT, R3                                        ; 1801
                             0B 18   00009          BGEQ    1$
                          7E  00G 8F   9A 0000B       MOVZBL  #BAS$K_ILLIO_CHA, -(SP)
              00000000G   00     01   FB 0000F       CALLS   #1, BAS$$STOP
                          7E  0C A2   7D 00016 1$:    MOVQ    12(FMP), -(SP)                                  ; 1809
                             53 D5   0001A          TSTL    R3
                             05 12   0001C          BNEQ    2$
                          7E    07   CE 0001E       MNEGL   #7, -(SP)
                             02 11   00021          BRB     3$
                             53 DD   00023 2$:      PUSHL   R3
                             20 DD   00025 3$:      PUSHL   #32
                    0000V    CF     04 FB 00027    CALLS   #4, BAS$$IO_BEG
                                    04 0002C       RET                                                     ; 1811

; Routine Size:  45 bytes,    Routine Base:  _BAS$CODE + 0082

;  519          1812  1
```

```
;  521          1813  1 GLOBAL ROUTINE BAS$INPUT (                    ! READ sequential list-directed
;  522          1814  1         UNIT                                  ! logical unit
;  523          1815  1     ) : NOVALUE =
;  524          1816  1
;  525          1817  1 !++
;  526          1818  1 ! ABSTRACT:
;  527          1819  1 !
;  528          1820  1 !       Initialize the BASIC I/O system to perform
;  529          1821  1 !       a READ sequential list-directed.
;  530          1822  1 !
;  531          1823  1 ! FORMAL PAREMETERS:
;  532          1824  1 !
;  533          1825  1 !       UNIT.rl.v       logical unit number
;  534          1826  1 !
;  535          1827  1 ! IMPLICIT INPUTS:
;  536          1828  1 !
;  537          1829  1 !       NONE
;  538          1830  1 !
;  539          1831  1 ! IMPLICIT OUTPUTS:
;  540          1832  1 !
;  541          1833  1 !       NONE
;  542          1834  1 !
;  543          1835  1 ! ROUTINE VALUE:
;  544          1836  1 !
;  545          1837  1 !       NONE
;  546          1838  1 !
;  547          1839  1 ! SIDE EFFECTS:
;  548          1840  1 !
;  549          1841  1 !       LUB/ISB/RAB control blocks allocated and file
;  550          1842  1 !       opened if not already. SIGNAL_STOPs many errors.
;  551          1843  1 !--
;  552          1844  1
;  553          1845  2     BEGIN
;  554          1846  2
;  555          1847  2     BUILTIN
;  556          1848  2         FP;
;  557          1849  2
;  558          1850  2     LOCAL
;  559          1851  2         FMP : REF BLOCK [, BYTE];
;  560          1852  2
;  561          1853  2     FMP = .FP;
;  562          1854  2 !+
;  563          1855  2 ! If channel number is less than zero then it is an illegal channel.
;  564          1856  2 !-
;  565          1857  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  566          1858  2
;  567          1859  2 !+
;  568          1860  2 ! Perform default OPEN if UNIT not already opened.
;  569          1861  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  570          1862  2 ! any errors.  Store I/O statement type code. Call
;  571          1863  2 ! UDF level initialization.
;  572          1864  2 ! If the unit number is 0, then change it.
;  573          1865  2 !-
;  574          1866  2     BAS$$IO_BEG (ISB$K_ST_TY_INP, (IF (.UNIT EQL 0) THEN LUB$K_LUN_INPU ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  575          1867  2         .FMP [SF$L_SAVE_PC]);
;  576          1868  1     END;
```

```
                                    000C 00000          .ENTRY  BAS$INPUT, Save R2,R3                   ; 1813
                            52      5D   D0 00002          MOVL    FP, FMP                                 ; 1853
                            53  04  AC   D0 00005          MOVL    UNIT, R3                                ; 1857
                                    0B   18 00009          BGEQ    1$                                      ;
                            7E  00G 8F   9A 0000B          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                 ;
                00000000G   00      01   FB 0000F          CALLS   #1, BAS$$STOP                           ;
                            7E  0C  A2   7D 00016 1$:      MOVQ    12(FMP), -(SP)                          ; 1866
                                    53   D5 0001A          TSTL    R3                                      ;
                                    05   12 0001C          BNEQ    2$                                      ;
                            7E      07   CE 0001E          MNEGL   #7, -(SP)                               ;
                                    02   11 00021          BRB     3$                                      ;
                                    53   DD 00023 2$:      PUSHL   R3                                      ;
                                    1E   DD 00025 3$:      PUSHL   #30                                     ;
                    0000V   CF      04   FB 00027          CALLS   #4, BAS$$IO_BEG                         ;
                                         04 0002C          RET                                             ; 1868

; Routine Size:  45 bytes,    Routine Base:  _BAS$CODE + 00AF

;  577          1869  1
```

```
;  579     1870  1 GLOBAL ROUTINE BAS$ANSI_INPUT (                    ! READ sequential list-directed
;  580     1871  1         UNIT                                       ! logical unit
;  581     1872  1     ) : NOVALUE =
;  582     1873  1
;  583     1874  1 !++
;  584     1875  1 ! ABSTRACT:
;  585     1876  1 !
;  586     1877  1 !       Initialize the BASIC I/O system to perform
;  587     1878  1 !       a READ sequential list-directed.  The only
;  588     1879  1 !       difference between this entry point and BAS$INPUT
;  589     1880  1 !       is that this one sets an ANSI flag in the LUB for
;  590     1881  1 !       later error processing.
;  591     1882  1 !
;  592     1883  1 ! FORMAL PAREMETERS:
;  593     1884  1 !
;  594     1885  1 !       UNIT.rl.v       logical unit number
;  595     1886  1 !
;  596     1887  1 ! IMPLICIT INPUTS:
;  597     1888  1 !
;  598     1889  1 !       NONE
;  599     1890  1 !
;  600     1891  1 ! IMPLICIT OUTPUTS:
;  601     1892  1 !
;  602     1893  1 !       NONE
;  603     1894  1 !
;  604     1895  1 ! ROUTINE VALUE:
;  605     1896  1 !
;  606     1897  1 !       NONE
;  607     1898  1 !
;  608     1899  1 ! SIDE EFFECTS:
;  609     1900  1 !
;  610     1901  1 !       LUB/ISB/RAB control blocks allocated and file
;  611     1902  1 !       opened if not already. SIGNAL_STOPs many errors.
;  612     1903  1 !--
;  613     1904  1
;  614     1905  2     BEGIN
;  615     1906  2
;  616     1907  2     BUILTIN
;  617     1908  2         FP;
;  618     1909  2
;  619     1910  2     LOCAL
;  620     1911  2         FMP : REF BLOCK [, BYTE];
;  621     1912  2
;  622     1913  2     FMP = .FP;
;  623     1914  2 !+
;  624     1915  2 ! If channel number is less than zero then it is an illegal channel.
;  625     1916  2 !-
;  626     1917  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  627     1918  2
;  628     1919  2 !+
;  629     1920  2 ! Perform default OPEN if UNIT not already opened.
;  630     1921  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  631     1922  2 ! any errors.  Store I/O statement type code. Call
;  632     1923  2 ! UDF level initialization.
;  633     1924  2 ! If the unit number is 0, then change it.
;  634     1925  2 !-
;  635     1926  2     BAS$$IO_BEG (ISB$K_ST_TY_INP, (IF (.UNIT EQL 0) THEN LUB$K_LUN_INPU ELSE .UNIT), .FMP [SF$L_SAVE_FP],
```

```
;  636          1927  2          .FMP [SF$L_SAVE_PC], 0, K_ANSI_FLAG);
;  637          1928  1      END;


                                    000C 00000          .ENTRY  BAS$ANSI_INPUT, Save R2,R3                  ; 1870
                            52      5D   D0 00002          MOVL    FP, FMP                                     ; 1913
                            53  04  AC   D0 00005          MOVL    UNIT, R3                                    ; 1917
                                    0B   18 00009          BGEQ    1$
                            7E  00G 8F   9A 0000B          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)
                00000000G   00      01   FB 0000F          CALLS   #1, BAS$$STOP
                                    02   DD 00016 1$:      PUSHL   #2                                          ; 1926
                                    7E   D4 00018          CLRL    -(SP)
                            7E  0C  A2   7D 0001A          MOVQ    12(FMP), -(SP)
                                    53   D5 0001E          TSTL    R3
                                    05   12 00020          BNEQ    2$
                            7E      07   CE 00022          MNEGL   #7, -(SP)
                                    02   11 00025          BRB     3$
                                    53   DD 00027 2$:      PUSHL   R3
                                    1E   DD 00029 3$:      PUSHL   #30
                    0000V   CF      06   FB 0002B          CALLS   #6, BAS$$IO_BEG
                                         04 00030          RET                                                 ; 1928

; Routine Size:  49 bytes,    Routine Base:  _BAS$CODE + 00DC

;  638          1929  1
```

```
;  640          1930  1 GLOBAL ROUTINE BAS$MAT_INPUT (                    ! MAT INPUT
;  641          1931  1         UNIT                                       ! logical unit
;  642          1932  1     ) : NOVALUE =
;  643          1933  1
;  644          1934  1 !++
;  645          1935  1 ! ABSTRACT:
;  646          1936  1 !
;  647          1937  1 !       Initialize the BASIC I/O system to perform
;  648          1938  1 !       the input of an array.  The array is named in the element transmitter.
;  649          1939  1 !
;  650          1940  1 ! FORMAL PAREMETERS:
;  651          1941  1 !
;  652          1942  1 !       UNIT.rl.v       logical unit number
;  653          1943  1 !
;  654          1944  1 ! IMPLICIT INPUTS:
;  655          1945  1 !
;  656          1946  1 !       NONE
;  657          1947  1 !
;  658          1948  1 ! IMPLICIT OUTPUTS:
;  659          1949  1 !
;  660          1950  1 !       NONE
;  661          1951  1 !
;  662          1952  1 ! ROUTINE VALUE:
;  663          1953  1 !
;  664          1954  1 !       NONE
;  665          1955  1 !
;  666          1956  1 ! SIDE EFFECTS:
;  667          1957  1 !
;  668          1958  1 !       LUB/ISB/RAB control blocks allocated and file
;  669          1959  1 !       opened if not already. SIGNAL_STOPs many errors.
;  670          1960  1 !--
;  671          1961  1
;  672          1962  2     BEGIN
;  673          1963  2
;  674          1964  2     BUILTIN
;  675          1965  2         FP;
;  676          1966  2
;  677          1967  2     LOCAL
;  678          1968  2         FMP : REF BLOCK [, BYTE];
;  679          1969  2
;  680          1970  2     FMP = .FP;
;  681          1971  2 !+
;  682          1972  2 ! If channel number is less than zero then it is an illegal channel.
;  683          1973  2 !-
;  684          1974  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  685          1975  2
;  686          1976  2 !+
;  687          1977  2 ! Perform default OPEN if UNIT not already opened.
;  688          1978  2 ! Check for  improper mixing of I/O statements. SIGNAL_STOP
;  689          1979  2 ! any errors.  Store I/O statement type code. Call
;  690          1980  2 ! UDF level initialization.
;  691          1981  2 ! If the unit number is 0, then change it.
;  692          1982  2 !-
;  693          1983  2     BAS$$IO_BEG (ISB$K_ST_TY_MIN, (IF (.UNIT EQL 0) THEN LUB$K_LUN_INPU ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  694          1984  2         .FMP [SF$L_SAVE_PC]);
;  695          1985  1     END;                                           ! End of BAS$MAT_INPUT
```

```
                                    000C 00000          .ENTRY  BAS$MAT_INPUT, Save R2,R3                       ; 1930
                            52      5D   D0 00002      MOVL    FP, FMP                                         ; 1970
                            53  04  AC   D0 00005      MOVL    UNIT, R3                                        ; 1974
                                0B  18   00009          BGEQ    1$
                            7E  00G 8F   9A 0000B      MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                         ;
                00000000G   00      01   FB 0000F      CALLS   #1, BAS$$STOP                                   ;
                            7E  0C  A2   7D 00016 1$:  MOVQ    12(FMP), -(SP)                                  ; 1983
                                    53   D5 0001A      TSTL    R3
                                    05   12 0001C      BNEQ    2$
                            7E      07   CE 0001E      MNEGL   #7, -(SP)
                                    02   11 00021      BRB     3$
                                    53   DD 00023 2$:  PUSHL   R3
                                    30   DD 00025 3$:  PUSHL   #48
                      0000V CF          04   FB 00027      CALLS   #4, BAS$$IO_BEG                             ;
                                        04 0002C          RET                                                     ; 1985

; Routine Size:  45 bytes,    Routine Base:  _BAS$CODE + 010D

;  696          1986  1
```

```
;  698          1987  1 GLOBAL ROUTINE BAS$PRINT (                    ! WRITE sequential list-directed
;  699          1988  1         UNIT                                   ! logical unit
;  700          1989  1     ) : NOVALUE =
;  701          1990  1
;  702          1991  1 !++
;  703          1992  1 ! ABSTRACT:
;  704          1993  1 !
;  705          1994  1 !       Initialize the BASIC I/O system to perform
;  706          1995  1 !       a PRINT sequential list-directed.
;  707          1996  1 !
;  708          1997  1 ! FORMAL PAREMETERS:
;  709          1998  1 !
;  710          1999  1 !       UNIT.rl.v       logical unit number
;  711          2000  1 !
;  712          2001  1 ! IMPLICIT INPUTS:
;  713          2002  1 !
;  714          2003  1 !       NONE
;  715          2004  1 !
;  716          2005  1 ! IMPLICIT OUTPUTS:
;  717          2006  1 !
;  718          2007  1 !       NONE
;  719          2008  1 !
;  720          2009  1 ! ROUTINE VALUE:
;  721          2010  1 !
;  722          2011  1 !       NONE
;  723          2012  1 !
;  724          2013  1 ! SIDE EFFECTS:
;  725          2014  1 !
;  726          2015  1 !       LUB/ISB/RAB control blocks allocated and file
;  727          2016  1 !       opened if not already. SIGNAL_STOPs many errors
;  728          2017  1 !--
;  729          2018  1
;  730          2019  2     BEGIN
;  731          2020  2
;  732          2021  2     BUILTIN
;  733          2022  2         FP;
;  734          2023  2
;  735          2024  2     LOCAL
;  736          2025  2         FMP : REF BLOCK [, BYTE];
;  737          2026  2
;  738          2027  2     FMP = .FP;
;  739          2028  2 !+
;  740          2029  2 ! If channel number is less than zero then it is an illegal channel.
;  741          2030  2 !-
;  742          2031  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  743          2032  2
;  744          2033  2     BAS$$IO_BEG (ISB$K_ST_TY_PRI, (IF (.UNIT EQL 0) THEN LUB$K_LUN_BPRI ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  745          2034  2         .FMP [SF$L_SAVE_PC]);
;  746          2035  1     END;                                        ! End of BAS$PRINT

                                000C 00000          .ENTRY  BAS$PRINT, Save R2,R3                   ; 1987
                            52  5D  D0 00002        MOVL    FP, FMP                                 ; 2027
                            53  04  AC  D0 00005    MOVL    UNIT, R3                                ; 2031
```

```
                                 0B   18 00009          BGEQ    1$                          ;
                7E         00G   8F   9A 0000B          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)     ;
       00000000G 00              01   FB 0000F          CALLS   #1, BAS$$STOP               ;
                7E          0C   A2   7D 00016 1$:      MOVQ    12(FMP), -(SP)              ; 2033
                                 53   D5 0001A          TSTL    R3                          ;
                                 05   12 0001C          BNEQ    2$                          ;
                7E               08   CE 0001E          MNEGL   #8, -(SP)                   ;
                                 02   11 00021          BRB     3$                          ;
                                 53   DD 00023 2$:      PUSHL   R3                          ;
                                 1B   DD 00025 3$:      PUSHL   #27                         ;
          0000V CF               04   FB 00027          CALLS   #4, BAS$$IO_BEG             ;
                                      04 0002C          RET                                 ; 2035

; Routine Size:  45 bytes,    Routine Base:  _BAS$CODE + 013A


;  747          2036  1
```

```
;  749          2037  1 GLOBAL ROUTINE BAS$ANSI_PRINT (                      ! WRITE sequential list-directed
;  750          2038  1                                                      ! to ANSI standards
;  751          2039  1         UNIT                                         ! logical unit
;  752          2040  1     ) : NOVALUE =
;  753          2041  1
;  754          2042  1 !++
;  755          2043  1 ! ABSTRACT:
;  756          2044  1 !
;  757          2045  1 !       Initialize the BASIC I/O system to perform
;  758          2046  1 !       a PRINT sequential list-directed.  The only
;  759          2047  1 !       difference between this entry point and BAS$PRINT
;  760          2048  1 !       is that this one sets an ANSI flag in the LUB for
;  761          2049  1 !       later error processing.
;  762          2050  1 !
;  763          2051  1 ! FORMAL PAREMETERS:
;  764          2052  1 !
;  765          2053  1 !       UNIT.rl.v       logical unit number
;  766          2054  1 !
;  767          2055  1 ! IMPLICIT INPUTS:
;  768          2056  1 !
;  769          2057  1 !       NONE
;  770          2058  1 !
;  771          2059  1 ! IMPLICIT OUTPUTS:
;  772          2060  1 !
;  773          2061  1 !       NONE
;  774          2062  1 !
;  775          2063  1 ! ROUTINE VALUE:
;  776          2064  1 !
;  777          2065  1 !       NONE
;  778          2066  1 !
;  779          2067  1 ! SIDE EFFECTS:
;  780          2068  1 !
;  781          2069  1 !       LUB/ISB/RAB control blocks allocated and file
;  782          2070  1 !       opened if not already. SIGNAL_STOPs many errors
;  783          2071  1 !--
;  784          2072  1
;  785          2073  2     BEGIN
;  786          2074  2
;  787          2075  2     BUILTIN
;  788          2076  2         FP;
;  789          2077  2
;  790          2078  2     LOCAL
;  791          2079  2         FMP : REF BLOCK [, BYTE];
;  792          2080  2
;  793          2081  2     FMP = .FP;
;  794          2082  2 !+
;  795          2083  2 ! If channel number is less than zero then it is an illegal channel.
;  796          2084  2 !-
;  797          2085  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  798          2086  2
;  799          2087  2     BAS$$IO_BEG (ISB$K_ST_TY_PRI, (IF (.UNIT EQL 0) THEN LUB$K_LUN_BPRI ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  800          2088  2         .FMP [SF$L_SAVE_PC], 0, K_ANSI_FLAG);
;  801          2089  1     END;                                              ! End of BAS$ANSI_PRINT
```

```
                                 000C 00000          .ENTRY  BAS$ANSI_PRINT, Save R2,R3                      ; 2037
                          52  5D   D0 00002          MOVL    FP, FMP                                         ; 2081
                          53  04 AC   D0 00005          MOVL    UNIT, R3                                        ; 2085
                              0B   18 00009          BGEQ    1$                                              :
                          7E  00G 8F   9A 0000B          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                         :
                00000000G 00      01   FB 0000F          CALLS   #1, BAS$$STOP                                   :
                                  02   DD 00016 1$:      PUSHL   #2                                              ; 2087
                                  7E   D4 00018          CLRL    -(SP)                                           :
                          7E  0C  A2   7D 0001A          MOVQ    12(FMP), -(SP)                                  :
                                  53   D5 0001E          TSTL    R3                                              :
                                  05   12 00020          BNEQ    2$                                              :
                          7E      08   CE 00022          MNEGL   #8, -(SP)                                       :
                                  02   11 00025          BRB     3$                                              :
                                  53   DD 00027 2$:      PUSHL   R3                                              :
                                  1B   DD 00029 3$:      PUSHL   #27                                             :
                     0000V CF           06   FB 0002B          CALLS   #6, BAS$$IO_BEG                                 :
                                       04 00030          RET                                                     ; 2089

; Routine Size:  49 bytes,    Routine Base:  _BAS$CODE + 0167

;  802          2090  1
```

```
;  804          2091  1 GLOBAL ROUTINE BAS$MAT_PRINT (                       ! MAT PRINT
;  805          2092  1         UNIT,                                        ! logical unit
;  806          2093  1         UPPER_BOUND1,                                ! No. of columns
;  807          2094  1         UPPER_BOUND2                                 ! No. of rows
;  808          2095  1     ) : NOVALUE =
;  809          2096  1
;  810          2097  1 !++
;  811          2098  1 ! ABSTRACT:
;  812          2099  1 !
;  813          2100  1 !       Initialize the BASIC I/O system to perform
;  814          2101  1 !       the printing of a matrix
;  815          2102  1 !
;  816          2103  1 ! FORMAL PAREMETERS:
;  817          2104  1 !
;  818          2105  1 !       UNIT.rl.v               logical unit number
;  819          2106  1 !       UPPER_BOUND1.rl.v       no. of columns
;  820          2107  1 !       UPPER_BOUND2.rl.v       no. of rows
;  821          2108  1 !
;  822          2109  1 ! IMPLICIT INPUTS:
;  823          2110  1 !
;  824          2111  1 !       NONE
;  825          2112  1 !
;  826          2113  1 ! IMPLICIT OUTPUTS:
;  827          2114  1 !
;  828          2115  1 !       NONE
;  829          2116  1 !
;  830          2117  1 ! ROUTINE VALUE:
;  831          2118  1 !
;  832          2119  1 !       NONE
;  833          2120  1 !
;  834          2121  1 ! SIDE EFFECTS:
;  835          2122  1 !
;  836          2123  1 !       LUB/ISB/RAB control blocks allocated and file
;  837          2124  1 !       opened if not already. SIGNAL_STOPs many errors
;  838          2125  1 !--
;  839          2126  1
;  840          2127  2     BEGIN
;  841          2128  2
;  842          2129  2     BUILTIN
;  843          2130  2         FP;
;  844          2131  2
;  845          2132  2     LOCAL
;  846          2133  2         FMP : REF BLOCK [, BYTE];
;  847          2134  2
;  848          2135  2     FMP = .FP;
;  849          2136  2 !+
;  850          2137  2 ! If channel number is less than zero then it is an illegal channel.
;  851          2138  2 !-
;  852          2139  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  853          2140  2
;  854          2141  2     BAS$$IO_BEG (ISB$K_ST_TY_MPR, (IF (.UNIT EQL 0) THEN LUB$K_LUN_BPRI ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  855          2142  2         .FMP [SF$L_SAVE_P]);
;  856          2143  1     END;                                              ! End of BAS$MAT_PRINT
```

```
                                 000C 00000          .ENTRY  BAS$MAT_PRINT, Save R2,R3                       ; 2091
                          52       5D  D0 00002      MOVL    FP, FMP                                         ; 2135
                          53  04   AC  D0 00005      MOVL    UNIT, R3                                        ; 2139
                              0B   18  18 00009      BGEQ    1$
                          7E  00G  8F  9A 0000B      MOVZBL  #BAS$K_ILLIO_CHA, -(SP)
                00000000G 00       01  FB 0000F      CALLS   #1, BAS$$STOP
                          7E  0C   A2  7D 00016 1$:  MOVQ    12(FMP), -(SP)                                  ; 2141
                              53   D5  0001A        TSTL    R3
                              05   12  0001C        BNEQ    2$
                          7E       08  CE 0001E      MNEGL   #8, -(SP)
                              02   11  00021        BRB     3$
                              53   DD  00023 2$:    PUSHL   R3
                              35   DD  00025 3$:    PUSHL   #53
                     0000V    CF   04  FB 00027      CALLS   #4, BAS$$IO_BEG
                                   04  0002C        RET                                                     ; 2143

; Routine Size:  45 bytes,    Routine Base:  _BAS$CODE + 0198

;  857          2144  1
```

```
;  859          2145  1 GLOBAL ROUTINE BAS$PRINT_USING (                  ! WRITE formatted
;  860          2146  1         UNIT,                                     ! logical unit
;  861          2147  1         FORMAT_STRING                             ! format string
;  862          2148  1 !       CURRENCY,                                 ! currency symbol for this stmt
;  863          2149  1 !       SEPARATOR,                                ! digit group separator
;  864          2150  1 !       DECIMAL_POINT                             ! decimal point
;  865          2151  1     ) : NOVALUE =
;  866          2152  1
;  867          2153  1 !++
;  868          2154  1 ! ABSTRACT:
;  869          2155  1 !
;  870          2156  1 !       Initialize the BASIC I/O system to perform
;  871          2157  1 !       a PRINT USING formatted output statement.
;  872          2158  1 !
;  873          2159  1 ! FORMAL PAREMETERS:
;  874          2160  1 !
;  875          2161  1 !       UNIT.rl.v       logical unit number
;  876          2162  1 !       FORMAT_STR.rt.dx        format string
;  877          2163  1 !       CURRENCY.rt.dx          optional currency symbol
;  878          2164  1 !       SEPARATOR.rt.dx         optional digit group separator symbol
;  879          2165  1 !       DECIMAL_POINT.rt.dx     optional decimal point symbol
;  880          2166  1 !
;  881          2167  1 ! IMPLICIT INPUTS:
;  882          2168  1 !
;  883          2169  1 !       NONE
;  884          2170  1 !
;  885          2171  1 ! IMPLICIT OUTPUTS:
;  886          2172  1 !
;  887          2173  1 !       NONE
;  888          2174  1 !
;  889          2175  1 ! ROUTINE VALUE:
;  890          2176  1 !
;  891          2177  1 !       NONE
;  892          2178  1 !
;  893          2179  1 ! SIDE EFFECTS:
;  894          2180  1 !
;  895          2181  1 !       LUB/ISB/RAB control blocks allocated and file
;  896          2182  1 !       opened if not already. SIGNAL_STOPs many errors
;  897          2183  1 !--
;  898          2184  1
;  899          2185  2     BEGIN
;  900          2186  2
;  901          2187  2     BUILTIN
;  902          2188  2         ACTUALCOUNT,
;  903          2189  2         FP;
;  904          2190  2
;  905          2191  2     LOCAL
;  906          2192  2         FMP : REF BLOCK [, BYTE];
;  907          2193  2
;  908          2194  2     FMP = .FP;
;  909          2195  2
;  910          2196  2 !+
;  911          2197  2 ! If channel number is less than zero then it is an illegal channel.
;  912          2198  2 !-
;  913          2199  2     IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  914          2200  2
;  915          2201  2     !+
```

```
;  916          2202  2     ! Call the routine which will set up the I/O data base and initialize a
;  917          2203  2     ! few areas
;  918          2204  2     !-
;  919          2205  2
;  920          2206  2     BAS$$IO_BEG (ISB$K_ST_TY_PRU, (IF (.UNIT EQL 0) THEN LUB$K_LUN_BPRI ELSE .UNIT), .FMP [SF$L_SAVE_FP],
;  921          2207  2         .FMP [SF$L_SAVE_PC], .FORMAT_STRING);
;  922          2208  2
;  923          2209  2     !+
;  924          2210  2     ! Now set a few Print Using specific areas in the ISB.  These are the currency
;  925          2211  2     ! symbol, the digit group separator, and the decimal point.
;  926          2212  2     !-
;  927          2213  2
;  928          2214  2 !    CCB = .OTS$$A_CUR_LUB;
;  929          2215  2 !    CASE .ACTUAL_COUNT() FROM K_NONE TO K_DECIMAL_POINT OF
;  930          2216  2 !    SET
;  931          2217  2 !    [K_NONE, OUTRANGE]:
;  932          2218  2 !        ;
;  933          2219  2 !    [K_CURRENCY]:
;  934          2220  2 !        CCB[ISB$] = .CURRENCY;
;  935          2221  2 !    [K_SEPARATOR]:
;  936          2222  2 !        BEGIN
;  937          2223  2 !        CCB[ISB$] = .CURRENCY;
;  938          2224  2 !        CCB[ISB$] = .SEPARATOR;
;  939          2225  2 !        END;
;  940          2226  2 !    [K_DECIMAL_POINT]:
;  941          2227  2 !        BEGIN
;  942          2228  2 !        CCB[ISB$] = .CURRENCY;
;  943          2229  2 !        CCB[ISB$] = .SEPARATOR;
;  944          2230  2 !        CCB[ISB$] = .DECIMAL_POINT;
;  945          2231  2 !        END;
;  946          2232  2 !    TES;
;  947          2233  2     RETURN;
;  948          2234  1     END;                                    ! End of routine BAS$PRINT_USING


                                        000C 00000          .ENTRY  BAS$PRINT_USING, Save R2,R3                 ; 2145
                              52        5D   D0 00002       MOVL    FP, FMP                                     ; 2194
                              53   04   AC   D0 00005       MOVL    UNIT, R3                                    ; 2199
                                   0B   18 00009       BGEQ    1$                                          ;
                              7E   00G  8F   9A 0000B       MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                     ;
                00000000G     00        01   FB 0000F       CALLS   #1, BAS$$STOP                               ;
                                   08   AC   DD 00016 1$:    PUSHL   FORMAT_STRING                               ; 2207
                              7E   0C   A2   7D 00019       MOVQ    12(FMP), -(SP)                              ; 2206
                                        53   D5 0001D       TSTL    R3                                          ;
                                        05   12 0001F       BNEQ    2$                                          ;
                              7E        08   CE 00021       MNEGL   #8, -(SP)                                   ;
                                        02   11 00024       BRB     3$                                          ;
                                        53   DD 00026 2$:    PUSHL   R3                                          ;
                                        1F   DD 00028 3$:    PUSHL   #31                                         ;
                     0000V      CF            05   FB 0002A       CALLS   #5, BAS$$IO_BEG                             ;
                                              04 0002F       RET                                                 ; 2234

; Routine Size:  48 bytes,    Routine Base:  _BAS$CODE + 01C5
```

```
;  949          2235  1
```

```
;  951          2236  1 GLOBAL ROUTINE BAS$$IO_BEG (                  ! Do common I/O statement initialization
;  952          2237  1         STTM_TYPE,                            ! Statement type - defined in ISB
;  953          2238  1         UNIT,                                ! Unit number (LUN) for this I/O statement
;  954          2239  1         FMP,                                 ! User's frame
;  955          2240  1         RESTART_PC,                          ! PC at which to restart I/O list
;  956          2241  1         FORMAT_STR,                          ! Format string (for PRINT USING)
;  957          2242  1         FLAGS                                ! Flags from INPUT, PRINT USING
;  958          2243  1     ) : NOVALUE =
;  959          2244  1 !+
;  960          2245  1 ! FUNCTIONAL DESCRIPTION:
;  961          2246  1 !
;  962          2247  1 !       Common I/O statement initialization:
;  963          2248  1 !
;  964          2249  1 !       1. Setup a LUB/ISB/RAB control block for this logical unit
;  965          2250  1 !          if not setup already.
;  966          2251  1 !       2. If unit not already OPEN, OPEN it.
;  967          2252  1 !       3. Check for incorrect mixing of I/O statements.
;  968          2253  1 !       4. Initialize values in the LUB/ISB/RAB
;  969          2254  1 !
;  970          2255  1 ! FORMAL PARAMETERS:
;  971          2256  1 !
;  972          2257  1 !       STTM_TYPE.rl.v          I/O statement type
;  973          2258  1 !       UNIT.rl.v               logical unit number
;  974          2259  1 !       FMP.ra.v                User's frame pointer, for getting major frame address
;  975          2260  1 !       RESTART_PC.ra.v         Pointer to I/O list code, for restarts.
;  976          2261  1 !       FORMAT_STR.rt.dx        Format string for write formatted
;  977          2262  1 !       [FLAGS.rlu.v]           Flags from INPUT or PRINT USING
;  978          2263  1 !                                1 = pre-compiled PRINT USING format string
;  979          2264  1 !                                2 = ANSI INPUT (special error checking required)
;  980          2265  1 !
;  981          2266  1 ! IMPLICIT INPUTS:
;  982          2267  1 !
;  983          2268  1 !       LUB$V_FORCIBLE          Flag indicating a forcible device.
;  984          2269  1 !       LUB$V_FORMATTED         This unit has been specified for
;  985          2270  1 !                               formatted I/O by a previous OPEN
;  986          2271  1 !                               or default OPEN.
;  987          2272  1 !       LUB$W_LUN               logical unit number, if unit
;  988          2273  1 !                               not already open
;  989          2274  1 !       LUB$V_OPENED            This unit has been opened by a previous
;  990          2275  1 !                               OPEN, or default OPEN (for READ/WRITE
;  991          2276  1 !                               OR ENDFILE).
;  992          2277  1 !       LUB$V_OUTBUF_DR         Flag to indicate that the output buffer has
;  993          2278  1 !                               valid data in it.
;  994          2279  1 !       LUB$V_READ_ONLY         This unit has been specified for
;  995          2280  1 !                               performing READs only by the current
;  996          2281  1 !                               OPEN or CALL FDBSET.
;  997          2282  1 !       LUB$V_UNFORMAT          This unit has been specified for
;  998          2283  1 !                               unformatted I/O by a previcus
;  999          2284  1 !                               OPEN, DEFINE FILE, or default OPEN.
; 1000          2285  1 !
; 1001          2286  1 ! IMPLICIT OUTPUTS:
; 1002          2287  1 !
; 1003          2288  1 !       ISB$B_SCALE_FAC         Scale factor in range of -6 -> 0.
; 1004          2289  1 !       ISB$B_STTM_TYPE         Statement type.  Used as an index into the UDF
; 1005          2290  1 !                               level dispatch tables.
; 1006          2291  1 !       ISB$V_DE_ENCODE         ENCODE/DECODE being done
; 1007          2292  1 !       LUB$A_BUF_PTR           Pointer to next character in buffer.
```

```
;1008     2293  1 !       LUB$A_BUF_END           Pointer to end+1 of buffer.
;1009     2294  1 !       LUB$L_LOG_RECNO         Current logical (or spanned)
;1010     2295  1 !                               record number for sequential access
;1011     2296  1 !                               files (needed for BACKSPACE of spanned
;1012     2297  1 !                               records). Current FORTRAN direct
;1013     2298  1 !                               access files 1 = first record.
;1014     2299  1 !                               0 never stored.
;1015     2300  1 !       ISB$B_ERR_NO            FORTRAN 0.  Last continuable error during statement
;1016     2301  1 !       LUB$V_IO_ACTIVE         Flag I/O active on this unit.
;1017     2302  1 !       ISB$W_FMT_LEN           If object-time format, no. of char.
;1018     2303  1 !                               allocated dynamically.  Needed for deallocation
;1019     2304  1 !       ISB$A_FMT_BEG           If object-time format, Adr. of first
;1020     2305  1 !                               char in resultant format array.
;1021     2306  1 !       ISB$A_RESTARTPC         PC at which to restart the I/O list after
;1022     2307  1 !                               certain errors.
;1023     2308  1 !
;1024     2309  1 ! ROUTINE VALUE:
;1025     2310  1 !
;1026     2311  1 !       NONE
;1027     2312  1 !
;1028     2313  1 ! SIDE EFFECTS:
;1029     2314  1 !
;1030     2315  1 !       Allocates LUB/ISB/RAB if not already allocated for
;1031     2316  1 !       this unit. Opens logical unit if not already open.
;1032     2317  1 !       Pushes down old current LUB/ISB/RAB (if any) and sets
;1033     2318  1 !       OTS$$A_CUR_LUB to new current LUB/ISB/RAB.
;1034     2319  1 !
;1035     2320  1 !       Signals:
;1036     2321  1 !       ILLEGAL OR ILLOGICAL ACCESS
;1037     2322  1 !       NOTE: An assumption is made that a Print statement that ends with a comma
;1038     2323  1 !       or a semicolon and that is followed by an Input statement is actually intended
;1039     2324  1 !       as a prompt.  The major implication is that there is not an error issued
;1040     2325  1 !       for trying to do output in the middle of input from a terminal format file
;1041     2326  1 !       (emphasis on file).  Instead the contents of the Print buffer are thrown
;1042     2327  1 !       away if we are not inputting from a terminal device.
;1043     2328  1 !
;1044     2329  1 !--
;1045     2330  1
;1046     2331  2     BEGIN
;1047     2332  2
;1048     2333  2     GLOBAL REGISTER
;1049     2334  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;1050     2335  2
;1051     2336  2     LITERAL
;1052     2337  2         K_FORMAT_STR = 5,                       ! Arg. No. of format string
;1053     2338  2         K_FLAGS_ARG = 6;                        ! Arg. No. of flags
;1054     2339  2
;1055     2340  2     BUILTIN
;1056     2341  2         ACTUALCOUNT;
;1057     2342  2
;1058     2343  2     MAP
;1059     2344  2         FMP : REF BLOCK [, BYTE] FIELD (BSF$FCD);
;1060     2345  2
;1061     2346  2     LOCAL
;1062     2347  2 !+
;1063     2348  2 ! Temp CCB for copying the contents of the Print
;1064     2349  2 ! buffer into the Prompt buffer if necessary
```

```
; 1065          2350  2 !-
; 1066          2351  2         TEMP_CCB : REF BLOCK [, BYTE];
; 1067          2352  2
; 1068          2353  2 !+
; 1069          2354  2 ! Allocate LUB/ISB/RAB for unit if not already setup
; 1070          2355  2 ! Push down active I/O.
; 1071          2356  2 ! Stores new LUB/ISB/RAB address in OTS Common OTS$$A_CUR_LUB.
; 1072          2357  2 ! If unit number out of range (LUB$K_ILUN_MIN:LUB$K_LUN_MAX)
; 1073          2358  2 ! SIGNAL BAS$_INVLOGUNI (32='INVALID LOGICAL UNIT NUMBER')
; 1074          2359  2 ! Store signed unit number in LUB$W_LUN.
; 1075          2360  2 !-
; 1076          2361  2     BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
; 1077          2362  2 !+
; 1078          2363  2 ! Now that the data base for this logical unit has been found, store off
; 1079          2364  2 ! the pointer to the most recent Basic major frame in the ISB.
; 1080          2365  2 ! Also store the restart address.
; 1081          2366  2 !-
; 1082          2367  2     CCB [ISB$A_RESTARTPC] = .RESTART_PC;
; 1083          2368  2     CCB [ISB$A_MAJ_F_PTR] =
; 1084          2369  3     BEGIN
; 1085          2370  3
; 1086          2371  3     IF (.FMP [BSF$A_HANDLER] EQLA BAS$HANDLER) THEN .FMP [BSF$A_BASE_R11] ELSE 0
; 1087          2372  3
; 1088          2373  2     END;
; 1089          2374  2     CCB [ISB$A_USER_FP] = .FMP;
; 1090          2375  2     CCB [ISB$B_STTM_TYPE] = .STTM_TYPE;
; 1091          2376  2 !+
; 1092          2377  2 ! Check for the optional flags argument.  For ANSI INPUT, the ANSI flag in
; 1093          2378  2 ! the LUB should be set so that the UDF and REC level routines will handle
; 1094          2379  2 ! errors differently.
; 1095          2380  2 !-
; 1096          2381  2     IF ACTUALCOUNT () EQL K_FLAGS_ARG
; 1097          2382  2     THEN
; 1098          2383  2         CCB [LUB$V_ANSI] = 1;
; 1099          2384  2 !+
; 1100          2385  2 ! Check for READONLY with an output operation or VIRTUAL usage.  Either error
; 1101          2386  2 ! signals ILLEGAL OR ILLOGICAL ACCESS.  Set the BLOCK USE bit.
; 1102          2387  2 !-
; 1103          2388  3         IF (.CCB [LUB$V_VA_USE] EQL 1)
; 1104          2389  3          OR (.CCB [LUB$V_READ_ONLY] AND .STTM_TYPE)
; 1105          2390  2         THEN
; 1106          2391  2             BAS$$STOP_IO(BAS$K_ILLILLACC);
; 1107          2392  2     CCB [LUB$V_BLK_USE] = 1;
; 1108          2393  2 !+
; 1109          2394  2 ! Check to see if there is a format string.
; 1110          2395  2 ! If there is, store the address away in two places in the ISB.  One copy is used
; 1111          2396  2 ! by the write formatted routines to keep their place in the format string; the
; 1112          2397  2 ! other is left pointing at the front of the string in case of reversion.
; 1113          2398  2 !-
; 1114          2399  2
; 1115          2400  2     IF ACTUALCOUNT () EQL K_FORMAT_STR
; 1116          2401  2     THEN
; 1117          2402  3         BEGIN
; 1118          2403  3
; 1119          2404  3         MAP
; 1120          2405  3             FORMAT_STR : REF BLOCK [8, BYTE];
; 1121          2406  3
```

```
;  1122          2407  3          CCB [ISB$A_FMT_PTR] = CCB [ISB$A_FMT_BEG] = .FORMAT_STR [DSC$A_POINTER];
;  1123          2408  3          CCB [ISB$W_LEN_REM] = CCB [ISB$W_FMT_LEN] = .FORMAT_STR [DSC$W_LENGTH];
;  1124          2409  2          END;
;  1125          2410  2
;  1126          2411  3      IF (.CCB [LUB$W_LUN] EQL LUB$K_LUN_BREAD)
;  1127          2412  2      THEN
;  1128          2413  3          BEGIN
;  1129          2414  3          CCB [LUB$V_FORMATTED] = 1;
;  1130          2415  3          CCB [ISB$V_DE_ENCODE] = 1;
;  1131          2416  3          END
;  1132          2417  3  !+
;  1133          2418  3  ! If unit not already open and #0 then OPEN it.
;  1134          2419  3  ! Otherwise, signal an error: UNOPENED FILE.
;  1135          2420  3  ! Set ACCESS depending on whether this is a sequential or
;  1136          2421  3  ! direct (.REC_NO > 0) access I/O statement
;  1137          2422  3  ! Set FORM depending on whether this is a formatted or
;  1138          2423  3  ! unformatted (.FORMAT_ADR = 0) I/O statement. Set
;  1139          2424  3  ! TYPE to 'NEW' or 'OLD' depending on whether this is
;  1140          2425  3  ! a write or read I/O statement. Note: Write statement type
;  1141          2426  3  ! codes are true, read are false. Any OPEN errors are
;  1142          2427  3  ! signaled.
;  1143          2428  3  !-
;  1144          2429  2      ELSE
;  1145          2430  3          BEGIN
;  1146          2431  3  !+
;  1147          2432  3  ! This is not a READ. Do an open default if not opened and unit 0.
;  1148          2433  3  ! Then check the opened flag and signal an error if it is not set.
;  1149          2434  3  !-
;  1150          2435  3
;  1151          2436  4          IF (( NOT .CCB [LUB$V_OPENED]) AND (.UNIT LSS 0))
;  1152          2437  3          THEN
;  1153          2438  3              BAS$$OPEN_ZERO (.FMP)
;  1154          2439  3          ELSE
;  1155          2440  3
;  1156          2441  4              IF (.UNIT GTR 0)
;  1157          2442  3              THEN
;  1158          2443  3  !+
;  1159          2444  3  ! Set LUB$A_BUDDY_PTR unconditionally for units > 0
;  1160          2445  3  !-
;  1161          2446  3                  CCB [LUB$A_BUDDY_PTR] = .CCB;
;  1162          2447  3
;  1163          2448  3  !+
;  1164          2449  3  ! If, after all that, the LUB is not open, then either it is not
;  1165          2450  3  ! channel zero or something went wrong in the OPEN code.  In either
;  1166          2451  3  ! case, indicate an error.
;  1167          2452  3  !-
;  1168          2453  3
;  1169          2454  3          IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  1170          2455  3
;  1171          2456  3  !+
;  1172          2457  3  ! These I/O statements are permitted only on a terminal format or sequential
;  1173          2458  3  ! file.
;  1174          2459  3  !-
;  1175          2460  3
;  1176          2461  3          IF (.CCB [LUB$V_NOTSEQORG]) THEN BAS$$STOP_IO (BAS$K_ILLRFORFIL);
;  1177          2462  3
;  1178          2463  2          END;
```

```
: 1179          2464  2 !+
: 1180          2465  2 ! Store off the scale factor.
: 1181          2466  2 !-
: 1182          2467  2     CCB [ISB$B_SCALE_FAC] = BAS$$SCALE_L_R1(.FMP);
: 1183          2468  2 !+
: 1184          2469  2 ! Check the Print buffer to see if there is valid data which has not been Put.
: 1185          2470  2 ! If there is valid data, then move it to the prompt buffer.  This is done
: 1186          2471  2 ! now rather than at the element transmitter UDF or REC level so that if
: 1187          2472  2 ! there is a prompt associated with this Input, it will be concatenated to
: 1188          2473  2 ! the contents of the Print buffer in the correct order.
: 1189          2474  2 ! We are able to look into LUB$A_BUDDY_PTR with great impugnity and not
: 1190          2475  2 ! worry about an access violation because if Print is not already opened,
: 1191          2476  2 ! Print is opened when Input is opened; we cannot get to this point unless Input
: 1192          2477  2 ! is open.
: 1193          2478  2 !-
: 1194          2479  2     TEMP_CCB = (IF (.CCB [LUB$A_BUDDY_PTR] NEQ 0) THEN .CCB [LUB$A_BUDDY_PTR] ELSE .CCB);
: 1195          2480  2
: 1196          2481  2 !+
: 1197          2482  2 ! If this is a INPUT or LINPUT or INPUT LINE then clear prompt buffer size
: 1198          2483  2 ! to prevent concatenation of prompts in case of ^C and resume.
: 1199          2484  2 ! Also zero print position of buddy unless previous PRINT terminator was
: 1200          2485  2 ! a semicolon or comma.
: 1201          2486  2 !-
: 1202          2487  2     IF .CCB [ISB$B_STTM_TYPE] EQLU ISB$K_ST_TY_LIN OR
: 1203          2488  2        .CCB [ISB$B_STTM_TYPE] EQLU ISB$K_ST_TY_INP OR
: 1204          2489  2        .CCB [ISB$B_STTM_TYPE] EQLU ISB$K_ST_TY_INL
: 1205          2490  2     THEN
: 1206          2491  3         BEGIN
: 1207          2492  3         CCB [RAB$B_PSZ] = 0;
: 1208          2493  4         TEMP_CCB [LUB$L_PRINT_POS] = (IF .TEMP_CCB [LUB$V_FORM_CHAR] EQLU 1
: 1209          2494  4                                       THEN .TEMP_CCB [LUB$L_PRINT_POS]
: 1210          2495  3                                       ELSE 0);
: 1211          2496  2         END;
: 1212          2497  2
: 1213          2498  4     IF (.TEMP_CCB [LUB$V_OUTBUF_DTR]) AND ((.CCB [ISB$B_STTM_TYPE] NEQ ISB$K_ST_TY_PRI) AND (.CCB [
: 1214          2499  3             ISB$B_STTM_TYPE] NEQ ISB$K_ST_TY_PRU))
: 1215          2500  2     THEN
: 1216          2501  3         BEGIN
: 1217          2502  3 !+
: 1218          2503  3 ! Check to see if this is a terminal device.  The contents of the Print
: 1219          2504  3 ! buffer are put into the prompt buffer only if this is a terminal device.
: 1220          2505  3 ! Otherwise, the contents of the Print buffer are thrown away.  Note that
: 1221          2506  3 ! if we just forced a Print at this point, an error would result because
: 1222          2507  3 ! we would be attempting to write in the middle of a sequential file.
: 1223          2508  3 !-
: 1224          2509  3
: 1225          2510  4         IF (.CCB [LUB$V_TERM_DEV])
: 1226          2511  3         THEN
: 1227          2512  4             BEGIN
: 1228          2513  4 !+
: 1229          2514  4 ! The Print buffer has valid data in it.  Concatenate it to the Prompt
: 1230          2515  4 ! buffer.  Clear the buffer dirty and format character flags.
: 1231          2516  4 ! Use OTS$$COPY instead of CH$COPY
: 1232          2517  4 !-
: 1233          2518  4             CH$COPY ((.TEMP_CCB [LUB$A_BUF_PTR] - .TEMP_CCB [LUB$A_BUF_BEG]),   ! Source element size
: 1234          2519  4                 ,TEMP_CCB [LUB$A_BUF_BEG],      ! Source address
: 1235          2520  4                 ' ',                            ! Fill character
```

```
;  1236          2521  4                 (.TEMP_CCB [LUB$A_BUF_PTR] - .TEMP_CCB [LUB$A_BUF_BEG]),        ! Destination element size
;  1237          2522  4                 .CCB [RAB$B_PSZ] + .CCB [RAB$L_PBF]);    ! Destination add-ess
;  1238          2523  4             CCB [RAB$B_PSZ] = .CCB [RAB$B_PSZ] + (.TEMP_CCB [LUB$A_BUF_PTR] - .TEMP_CCB [LUB$A_BUF_BEG]);
;  1239          2524  4             TEMP_CCB [LUB$V_OUTBUF_DR] = 0;
;  1240          2525  4             TEMP_CCB [LUB$V_FORM_CHAR] = 0;
;  1241          2526  4     !+
;  1242          2527  4     ! Reset the Print buffer pointer now that is not 'dirty'.
;  1243          2528  4     !-
;  1244          2529  4             TEMP_CCB [LUB$A_BUF_PTR] = .TEMP_CCB [LUB$A_BUF_BEG];
;  1245          2530  3                 END;
;  1246          2531  3
;  1247          2532  2             END;
;  1248          2533  2
;  1249          2534  2     !+
;  1250          2535  2     ! Call appropriate User data formatted level of abstraction
;  1251          2536  2     ! (UDF level = level 2) initialization routine.
;  1252          2537  2     !-
;  1253          2538  2         JSB_UDF0 (BAS$$AA_UDF_PR0 + .BAS$$AA_UDF_PR0 [.CCB [ISB$B_STTM_TYPE] - ISB$K_BASSTTYLO + 1]);
;  1254          2539  1         END;                                        ! End of BAS$$IO_BEG routine


                                    0FFC 00000          .ENTRY  BAS$$IO_BEG, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 2236
                                                                R10,R11
                              59 00000000G   00  9E 00002       MOVAB   BAS$$STOP_IO, R9
                              50             08  CE 00009       MNEGL   #8, R0                                ; 2361
                              52       08    AC  D0 0000C       MOVL    UNIT, R2
                                 00000000G   00  16 00010       JSB     BAS$$CB_PUSH
                         FF44 CB       10    AC  D0 00016       MOVL    RESTART_PC, -188(CCB)                 ; 2367
                              52       0C    AC  D0 0001C       MOVL    FMP, R2                               ; 2371
                              50 00000000G   00  9E 00020       MOVAB   BAS$HANDLER, R0
                              50             62  D1 00027       CMPL    (R2), R0
                                             06  12 0002A       BNEQ    1$
                              50       F4    A2  D0 0002C       MOVL    -12(R2), R0
                                             02  11 00030       BRB     2$
                              50             D4  00032 1$:      CLRL    R0
                         FF48 CB             50  D0 00034 2$:   MOVL    R0, -184(CCB)                         ; 2369
                         FF4C CB             52  D0 00039       MOVL    R2, -180(CCB)                         ; 2374
                              58       FF71  CB  9E 0003E       MOVAB   -143(CCB), R8                         ; 2375
                              68       04    AC  90 00043       MOVB    STTM_TYPE, (R8)
                              06             6C  91 00047       CMPB    (AP), #6                              ; 2381
                                             04  12 0004A       BNEQ    3$
                           A1 AB             10  88 0004C       BISB2   #16, -95(CCB)                         ; 2383
                              09       FF    AB  E8 00050 3$:   BLBS    -1(CCB), 4$                           ; 2388
                        0B    FC AB          02  E1 00054       BBC     #2, -4(CCB), 5$                       ; 2389
                              07       04    AC  E9 00059       BLBC    STTM_TYPE, 5$
                              7E       00G   8F  9A 0005D 4$:   MOVZBL  #BAS$K_ILLILLACC, -(SP)               ; 2391
                              69             01  FB 00061       CALLS   #1, BAS$$STOP_IO
                           FF AB             02  88 00064 5$:   BISB2   #2, -1(CCB)                          ; 2392
                              05             6C  91 00068       CMPB    (AP), #5                              ; 2400
                                             1D  12 0006B       BNEQ    6$
                              50       14    AC  D0 0006D       MOVL    FORMAT_STR, R0                        ; 2407
                              51       04    A0  D0 00071       MOVL    4(R0), R1
                         FF7C CB             51  D0 00075       MOVL    R1, -132(CCB)
                           80 AB             51  D0 0007A       MOVL    R1, -128(CCB)
```

```
                        50              60  3C  0007E          MOVZWL  (R0), R0                        ; 2408
                    FF72    CB          50  B0  00081          MOVW    R0, -142(CCB)
                      8D    AB          50  B0  00086          MOVW    R0, -115(CCB)
                    FFFA    8F      C6  AB  B1  0008A 6$:      CMPW    -58(CCB), #-6                   ; 2411
                                        0B  12  00090          BNEQ    7$
                      FD    AB          01  88  00092          BISB2   #1, -3(CCB)                     ; 2414
                      96    AB      40  8F  88  00096          BISB2   #64, -106(CCB)                  ; 2415
                                        34  11  0009B          BRB     11$                             ; 2411
                            10      FC  AB  E8  0009D 7$:      BLBS    -4(CCB), 8$                     ; 2436
                                    08  AC  D5  000A1          TSTL    UNIT
                                        08  18  000A4          BGEQ    8$
                                        52  DD  000A6          PUSHL   R2                              ; 2438
                00000000G           00      01  FB  000A8          CALLS   #1, BAS$$OPEN_ZERO
                                        09  11  000AF          BRB     9$
                                    08  AC  D5  000B1 8$:      TSTL    UNIT                            ; 2441
                                        04  15  000B4          BLEQ    9$
                      B8    AB          5B  D0  000B6          MOVL    CCB, -72(CCB)                   ; 2446
                            07      FC  AB  E8  000BA 9$:      BLBS    -4(CCB), 10$                    ; 2454
                            7E      00G 8F  9A  000BE          MOVZBL  #BAS$K_IO_CHANOT, -(SP)
                            69          01  FB  000C2          CALLS   #1, BAS$$STOP_IO
                07          A1  AB      03  E1  000C5 10$:     BBC     #3, -95(CCB), 11$               ; 2461
                            7E      00G 8F  9A  000CA          MOVZBL  #BAS$K_TERFORFIL, -(SP)
                            69          01  FB  000CE          CALLS   #1, BAS$$STOP_IO
                            50          52  D0  000D1 11$:     MOVL    R2, R0                          ; 2467
                                00000000G   00  16  000D4          JSB     BAS$$SCALE_L_R1
                    FF70    CB          50  90  000DA          MOVB    R0, -144(CCB)
                                B8  AB      D5  000DF          TSTL    -72(CCB)                        ; 2479
                                        06  13  000E2          BEQL    12$
                            56      B8  AB  D0  000E4          MOVL    -72(CCB), TEMP_CCB
                                        03  11  000E8          BRB     13$
                            56          5B  D0  000EA 12$:     MOVL    CCB, TEMP_CCB
                            1C          68  91  000ED 13$:     CMPB    (R8), #28                       ; 2487
                                        0A  13  000F0          BEQL    14$
                            1E          68  91  000F2          CMPB    (R8), #30                       ; 2488
                                        05  13  000F5          BEQL    14$
                            20          68  91  000F7          CMPB    (R8), #32                       ; 2489
                                        14  12  000FA          BNEQ    17$
                                34  AB  94  000FC 14$:      CLRB    52(CCB)                         ; 2492
                06          FE  A6      02  E1  000FF          BBC     #2, -2(TEMP_CCB), 15$           ; 2493
                            50      C8  A6  D0  00104          MOVL    -56(TEMP_CCB), R0               ; 2494
                                        02  11  00108          BRB     16$
                                        50  D4  0010A 15$:     CLRL    R0                              ; 2493
                            C8      A6      50  D0  0010C 16$:     MOVL    R0, -56(TEMP_CCB)
                2F          FE  A6      03  E1  00110 17$:     BBC     #3, -2(TEMP_CCB), 18$           ; 2498
                            1B          68  91  00115          CMPB    (R8), #27
                                        2A  13  00118          BEQL    18$
                            1F          68  91  0011A          CMPB    (R8), #31                       ; 2499
                                        25  13  0011D          BEQL    18$
                20          FE  AB      05  E1  0011F          BBC     #5, -2(CCB), 18$                ; 2510
                57          B0  A6  BC  A6  C3  00124          SUBL3   -68(TEMP_CCB), -80(TEMP_CCB), R7 ; 2518
                            50  34  AB  9A  0012A          MOVZBL  52(CCB), R0                     ; 2522
                            50  30  AB  C0  0012E          ADDL2   48(CCB), R0
                60          BC  B6      57  28  00132          MOVC3   R7, @-68(TEMP_CCB), (R0)
                            34  AB      57  80  00137          ADDB2   R7, 52(CCB)                     ; 2523
                            FE  A6      0C  8A  0013B          BICB2   #12, -2(TEMP_CCB)               ; 2525
                            B0  A6  BC  A6  D0  0013F          MOVL    -68(TEMP_CCB), -80(TEMP_CCB)    ; 2529
                            50          68  9A  00144 18$:     MOVZBL  (R8), R0                        ; 2538
```

```
                     50 0000000000G0040  DO 00147         MOVL    BAS$$AA_UDF_PRO-104[RO], RO      ;
                        0000000000G0040  16 0014F         JSB     BAS$$AA_UDF_PRO[RO]              ;
                                         04 00156         RET                                      ; 2539

; Routine Size:  343 bytes,    Routine Base:  _BAS$CODE + 01F5

; 1255          2540  1
; 1256          2541  1 END                                   ! End of BAS$IO_BEG module
; 1257          2542  1
; 1258          2543  0 ELUDOM
```

```
;                                PSECT SUMMARY
;
;      Name                     Bytes                          Attributes
;
;   _BAS$CODE                     844  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         6         0         581        00:01.2
```

```
;                                   COMMAND QUALIFIERS

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

; Size:         844 code + 0 data bytes
; Run Time:        00:26.3
; Elapsed Time:    01:01.9
; Lines/CPU Min:    5799
; Lexemes/CPU-Min: 27842
; Memory Used:  223 pages
; Compilation Complete
```

BASINIGSC
LIS

BASINIT
LIS

BASINIDEF
LIS

BASINIDFS
LIS

BASINIGSB
LIS

BASINSTR
LIS

BASINIONE
LIS

BASLEFT
LIS

BASMARGIN
LIS

BASINIIOL
LIS

BASKILL
LIS

BASTOBEG
LIS

BASTOEND
LIS

BASMATADD
LIS

BASMAGTAP
LIS