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

FILEID**BASGET

```
BBBBBBBB      AAAAAA      SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT
BBBBBBBB      AAAAAA      SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT
BB      BB   AA      AA  SS          GG          EE              TT
BB      BB   AA      AA  SS          GG          EE              TT
BB      BB   AA      AA  SS          GG          EE              TT
BB      BB   AA      AA  SS          GG          EE              TT
BBBBBBBB     AA      AA    SSSSSS    GG          EEEEEEEE        TT
BBBBBBBB     AA      AA    SSSSSS    GG          EEEEEEEE        TT
BB      BB   AAAAAAAAAA          SS  GG  GGGGGG  EE              TT
BB      BB   AAAAAAAAAA          SS  GG  GGGGGG  EE              TT
BB      BB   AA      AA          SS  GG      GG  EE              TT          ....
BB      BB   AA      AA          SS  GG      GG  EE              TT          ....
BBBBBBBB     AA      AA  SSSSSSSS      GGGGGG    EEEEEEEEEE      TT          ....
BBBBBBBB     AA      AA  SSSSSSSS      GGGGGG    EEEEEEEEEE      TT          ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE BAS$GET (                                  ! Basic GET construct
;     2          0002  0                 IDENT = '1-021'                  ! File: basget.b32 Edit:PLL1021
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !*****************************************************************************
;     7          0007  1 !*                                                                           *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                     *
;    11          0011  1 !*                                                                           *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                             *
;    18          0018  1 !*                                                                           *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                             *
;    22          0022  1 !*                                                                           *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*****************************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:
;    32          0032  1 !       Basic support library - user callable
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !       This module is the UPI level of the Basic GET construct.  Initially,
;    36          0036  1 !       it contains only the code for sequential I/O.  This module will set
;    37          0037  1 !       up the I/O data base for the LUN and dispatch to the UDF level.
;    38          0038  1 !
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !       User access mode - AST reentrant.
;    42          0042  1 !
;    43          0043  1 ! AUTHOR: Donald G. Petersen, CREATION DATE: 19-Feb-79
;    44          0044  1 !
;    45          0045  1 ! MODIFIED BY:
;    46          0046  1 !
;    47          0047  1 !       DGP, 19-Feb-79 : VERSION 01
;    48          0048  1 ! 1-001 - original.  DGP 19-Feb-79
;    49          0049  1 ! 1-002 - Put () after JSB to BAS$$REC_GSE so Bliss won't optimize it out.
;    50          0050  1 !         DGP 22-Feb-79
;    51          0051  1 ! 1-003 - Add BAS$GET_RECORD.  DGP 02-Mar-79
;    52          0052  1 ! 1-004 - More work on relative I/O.  DGP 05-Mar-79
;    53          0053  1 ! 1-005 - Add all of the trash for 'foreign buffers'.  DGP 26-Mar-79
;    54          0054  1 ! 1-006 - Make all external references use general addressing.  JBS 28-MAR-1979
;    55          0055  1 ! 1-007 - Remove library file RTLSTARLE, not used.  JBS 28-MAR-1979
;    56          0056  1 ! 1-008 - Load register CCB properly before second call to CB_POP.
;    57          0057  1 !         JBS 29-MAR-1979
```

```
;    58          0058  1 ! 1-009 - Add GET indexed.  DGP 03-Apr-79
;    59          0059  1 ! 1-010 - One too many arguments in call to BAS$$REC_GIN in BAS$GET_KEY.
;    60          0060  1 !         DGP 10-Apr-79
;    61          0061  1 ! 1-011 - Treat channel 0 correctly and check for channel not open.
;    62          0062  1 !         JBS 19-APR-1979
;    63          0063  1 ! 1-012 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
;    64          0064  1 ! 1-013 - Signal virtual array usage and set block use flag.  DGP 16-Oct-79
;    65          0065  1 ! 1-014 - Signal ILLIO_CHA if channel passed is less than zero. FM 10-sep-80
;    66          0066  1 ! 1-015 - Pass to BAS$$CB_PUSH, LUB$K_ILUN_MIN+2, as a result GET #0 BASIC
;    67          0067  1 !         statement will generate an error.  FM 17-SEP-80
;    68          0068  1 ! 1-016 - Undo 15.  We can now do I/O to #0, because BAS$PUT will now use
;    69          0069  1 !         foreign buffer mechanism to do #0 PUTs. FM 9-JUL-81.
;    70          0070  1 ! 1-017 - Fixed a couple of comments to reflect how channel 0 problem is fixed.
;    71          0071  1 !         FM 9-jul-81.
;    72          0072  1 ! 1-018 - Add support for RFA access and manual record locking.  PLL 1-Jun-82
;    73          0073  1 ! 1-019 - RFA is passed by ref, not descriptor.  PLL 4-Jun-1982
;    74          0074  1 ! 1-020 - Include RFA entry point in FORWARD.  PLL 9-Jun-1982
;    75          0075  1 ! 1-021 - Allow REGARDLESS (RRL bit) without UNLOCK EXPLICIT (ULK bit).  PLL 10-Jun-1982
;    76          0076  1 !--
;    77          0077  1
;    78          0078  1 !<BLF/PAGE>
```

```
;    80          0079  1 !
;    81          0080  1 ! SWITCHES:
;    82          0081  1 !
;    83          0082  1
;    84          0083  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    85          0084  1
;    86          0085  1 !
;    87          0086  1 ! LINKAGES
;    88          0087  1 !
;    89          0088  1
;    90          0089  1 REQUIRE 'RTLIN:OTSLNK';                         ! Define all linkages
;    91          0518  1
;    92          0519  1 !
;    93          0520  1 ! TABLE OF CONTENTS:
;    94          0521  1 !
;    95          0522  1
;    96          0523  1 FORWARD ROUTINE
;    97          0524  1     BAS$GET_RECORD : NOVALUE,                   ! UPI level Relative GET
;    98          0525  1     BAS$GET_KEY : NOVALUE,                      ! UPI level Indexed GET
;    99          0526  1     BAS$GET : NOVALUE,                          ! UPI level Sequential GET
;   100          0527  1     BAS$GET_RFA : NOVALUE;                      ! UPI level RFA GET
;   101          0528  1
;   102          0529  1 !
;   103          0530  1 ! INCLUDE FILES:
;   104          0531  1 !
;   105          0532  1
;   106          0533  1 REQUIRE 'RTLML:OTSISB';                         ! ISB definitions
;   107          0701  1
;   108          0702  1 REQUIRE 'RTLML:BASPAR';                         ! Basic literal for foreign buffer kludge
;   109          0724  1
;   110          0725  1 REQUIRE 'RTLML:OTSLUB';                         ! LUB definitions
;   111          0865  1
;   112          0866  1 REQUIRE 'RTLIN:RTLPSECT';                       ! Define DECLARE_PSECTS macro
;   113          0961  1
;   114          0962  1 LIBRARY 'RTLSTARLE';                            ! System symbols
;   115          0963  1
;   116          0964  1 !
;   117          0965  1 ! MACROS:
;   118          0966  1 !
;   119          0967  1 !       NONE
;   120          0968  1 !
;   121          0969  1 ! EQUATED SYMBOLS:
;   122          0970  1 !
;   123          0971  1 !       NONE
;   124          0972  1 !
;   125          0973  1 !
;   126          0974  1 ! PSECT DECLARATIONS:
;   127          0975  1 !
;   128          0976  1 DECLARE_PSECTS (BAS);
;   129          0977  1 !
;   130          0978  1 ! OWN STORAGE:
;   131          0979  1 !
;   132          0980  1 !       NONE
;   133          0981  1 !
;   134          0982  1 ! EXTERNAL REFERENCES:
;   135          0983  1 !
;   136          0984  1
```

```
;  137          0985  1 EXTERNAL ROUTINE
;  138          0986  1     BAS$$OPEN_ZERO;                          ! Open "channel 0"
;  139          0987  1
;  140          0988  1 EXTERNAL LITERAL
;  141          0989  1     BAS$K_IO_CHANOT : UNSIGNED (8),          ! I/O channel not open
;  142          0990  1     BAS$K_ILLILLACC : UNSIGNED (8),          ! Illegal or illogical access
;  143          0991  1     BAS$K_ILLIO_CHA : UNSIGNED (8),          ! Illegal I/O channel
;  144          0992  1     BAS$K_ILLRECLOC : UNSIGNED (8);          ! Illegal record locking clause
;  145          0993  1
;  146          0994  1 EXTERNAL ROUTINE
;  147          0995  1     BAS$$REC_GIN : JSB_REC_IND1 NOVALUE,     ! REC level - RMS interface, GET indexed
;  148          0996  1     BAS$$REC_GRE : JSB_DO_READ NOVALUE,      ! REC level - RMS interface GET relative
;  149          0997  1     BAS$$REC_GSE : JSB_DO_READ NOVALUE,      ! REC level processing - RMS interface
;  150          0998  1                                              ! GET sequential
;  151          0999  1     BAS$$REC_GRFA : JSB_DO_READ NOVALUE,     ! REC level - GET by RFA
;  152          1000  1     BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,      ! Push down I/O system
;  153          1001  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,        ! Pop I/O system back one CB
;  154          1002  1     BAS$$STOP_IO : NOVALUE,                  ! Signal fatal I/O errors
;  155          1003  1     BAS$$STOP : NOVALUE;                     ! Signal fatal BASIC error
;  156          1004  1
```

```
;  158          1005  1 GLOBAL ROUTINE BAS$GET (                      ! GET sequential
;  159          1006  1         UNIT,                                 ! logical unit number
;  160          1007  1         LOCK_FLAGS                            ! manual locking flagss
;  161          1008  1     ) : NOVALUE =
;  162          1009  1
;  163          1010  1 !++
;  164          1011  1 ! FUNCTIONAL DESCRIPTION:
;  165          1012  1 !
;  166          1013  1 !       This routine will set up the I/O data base for this LUN if necessary
;  167          1014  1 !       and then dispatch off to the REC level.  When control is returned to
;  168          1015  1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
;  169          1016  1 !       face to RMS is done at the REC level.  One record is read.
;  170          1017  1 !
;  171          1018  1 ! FORMAL PARAMETERS:
;  172          1019  1 !
;  173          1020  1 !       UNIT.rlu.v      logical unit number
;  174          1021  1 !       [LOCK_FLAGS.rlu.v]  if present, bits to pass on to record level to
;  175          1022  1 !                           control manual record locking
;  176          1023  1 !
;  177          1024  1 ! IMPLICIT INPUTS:
;  178          1025  1 !
;  179          1026  1 !       LUB$V_VA_USE    virtual array usage
;  180          1027  1 !
;  181          1028  1 ! IMPLICIT OUTPUTS:
;  182          1029  1 !
;  183          1030  1 !       OTS$$A_CUR_LUB          pointer to current control block
;  184          1031  1 !       RECOUNT                 Basic Global which contains the number of bytes read
;  185          1032  1 !       ISB$B_STTM_TYPE         the statement type
;  186          1033  1 !       LUB$V_BLK_USE           this file has been used for other than virtual I/O
;  187          1034  1 !
;  188          1035  1 ! COMPLETION CODES:
;  189          1036  1 !
;  190          1037  1 !       NONE
;  191          1038  1 !
;  192          1039  1 ! SIDE EFFECTS:
;  193          1040  1 !
;  194          1041  1 !       RECOUNT is assigned the number of bytes read.
;  195          1042  1 !       Signals:
;  196          1043  1 !       BAS$K_IO_CHANOT (I/O channel not open)
;  197          1044  1 !       BAS$K_ILLIO_CHA (illegal I/O channel) for foreign buffers.
;  198          1045  1 !
;  199          1046  1 !--
;  200          1047  1
;  201          1048  2     BEGIN
;  202          1049  2
;  203          1050  2     BUILTIN
;  204          1051  2         FP,
;  205          1052  2         ACTUALCOUNT;
;  206          1053  2
;  207          1054  2     LITERAL
;  208          1055  2         K_LOCK_ARG = 2;
;  209          1056  2
;  210          1057  2     GLOBAL REGISTER
;  211          1058  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  212          1059  2
;  213          1060  2     LOCAL
;  214          1061  2         FMP : REF BLOCK [, BYTE],
```

```
;  215          1062  2          ACTUAL_UNIT,                    ! Unit number, without foreign buffer
;  216          1063  2          TEMP_R11,                       ! CCB for foreign buffer, or 0
;  217          1064  2          FLAGS;
;  218          1065  2
;  219          1066  2  !+
;  220          1067  2  ! If channel is less than zero then signal an error.
;  221          1068  2  !-
;  222          1069  2      IF ( .UNIT LSS 0 ) THEN BAS$$STOP(BAS$K_ILLIO_CHA);
;  223          1070  2
;  224          1071  2      FMP = .FP;
;  225          1072  2  !+
;  226          1073  2  ! Check for "foreign buffers".  If the unit number exceeds 255 then a foreign
;  227          1074  2  ! buffer is specified.  The foreign buffer is actually a unit number whose
;  228          1075  2  ! buffer is to receive the record which is read.  The "foreign buffer" unit
;  229          1076  2  ! is pushed to pick up the CB address which is passed to the REC level.  Then
;  230          1077  2  ! the unit pointing to the file is pushed so that the CCB points to the log-
;  231          1078  2  ! ical unit which actually do the I/O.  Upon return, the necessary RAB fields
;  232          1079  2  ! (USZ and UBF) have been restored and two CB_POPs are done if necessary.
;  233          1080  2  !-
;  234          1081  2      TEMP_R11 = 0;
;  235          1082  2      ACTUAL_UNIT = .UNIT;
;  236          1083  2
;  237          1084  3      IF (.UNIT GTR LUB$K_LUN_MAX)
;  238          1085  2      THEN
;  239          1086  3          BEGIN
;  240          1087  3
;  241          1088  3          LOCAL
;  242          1089  3              FOREIGN_BUFFER;
;  243          1090  3
;  244          1091  3          FOREIGN_BUFFER = .UNIT/BAS$K_LUN_MAX;
;  245          1092  3          ACTUAL_UNIT = .UNIT MOD BAS$K_LUN_MAX;
;  246          1093  3
;  247          1094  3          IF (.FOREIGN_BUFFER GTRU BAS$K_MAX_FOR_B) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
;  248          1095  3
;  249          1096  3          BAS$$CB_PUSH (.FOREIGN_BUFFER, LUB$K_LUN_MIN);
;  250          1097  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  251          1098  3
;  252          1099  3          IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  253          1100  3
;  254          1101  3          TEMP_R11 = .CCB;
;  255          1102  2          END;
;  256          1103  2
;  257          1104  2  !+
;  258          1105  2  ! Worry about channel zero.  If the actual unit number is zero, make sure
;  259          1106  2  ! "channel 0" is open and use its input side so #0 I/O can work.
;  260          1107  2  !-
;  261          1108  2
;  262          1109  2      IF (.ACTUAL_UNIT EQL 0) THEN ACTUAL_UNIT = LUB$K_LUN_INPU;
;  263          1110  2
;  264          1111  2      BAS$$CB_PUSH (.ACTUAL_UNIT, LUB$K_ILUN_MIN);
;  265          1112  2      CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  266          1113  2  !+
;  267          1114  2  ! If we are on a default unit (unit number less than zero) then
;  268          1115  2  ! we can open it if it is not already open.  Otherwise it must
;  269          1116  2  ! be open already.
;  270          1117  2  !-
;  271          1118  2
```

```
;  272          1119  3     IF ( NOT .CCB [LUB$V_OPENED])
;  273          1120  2     THEN
;  274          1121  2
;  275          1122  3         IF (.ACTUAL_UNIT LSS 0)
;  276          1123  2         THEN
;  277          1124  3             BEGIN
;  278          1125  3             BAS$$OPEN_ZERO (.FMP [SF$L_SAVE_FP])
;  279          1126  3             END
;  280          1127  2         ELSE
;  281          1128  3             BEGIN
;  282          1129  3             BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  283          1130  2             END;
;  284          1131  2
;  285          1132  2 !+
;  286          1133  2 ! Now that the data base is in place, store the statement type and  go
;  287          1134  2 ! directly to the REC level.
;  288          1135  2 !-
;  289          1136  2     CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_GSE;
;  290          1137  2 !+
;  291          1138  2 ! Check for virtual array usage and set block usage.
;  292          1139  2 !-
;  293          1140  2     IF .CCB [LUB$V_VA_USE] EQL 1 THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  294          1141  2     CCB [LUB$V_BLK_USE] = 1;
;  295          1142  2
;  296          1143  2     IF ACTUALCOUNT () LSS K_LOCK_ARG
;  297          1144  2     THEN
;  298          1145  2         FLAGS = 0
;  299          1146  2     ELSE
;  300          1147  3         BEGIN
;  301          1148  3         !+
;  302          1149  3         ! The ULK bit must set unless this is a REGARDLESS clause.
;  303          1150  3         !-
;  304          1151  3         CASE .CCB [RAB$V_ULK] FROM 0 TO 1 OF
;  305          1152  3         SET
;  306          1153  3             [0]:
;  307          1154  3             IF (.LOCK_FLAGS AND RAB$M_RRL) NEQ 0
;  308          1155  3             THEN
;  309          1156  3                 FLAGS = .LOCK_FLAGS
;  310          1157  3             ELSE
;  311          1158  3                 BAS$$STOP_IO (BAS$K_ILLRECLOC);
;  312          1159  3
;  313          1160  3             [1]:
;  314          1161  3             FLAGS = .LOCK_FLAGS;
;  315          1162  3         TES;
;  316          1163  2         END;
;  317          1164  2     BAS$$REC_GSE (.TEMP_R11, .FLAGS);
;  318          1165  2 !+
;  319          1166  2 ! Now that the GET has been done, pop the CCB off the I/O system.
;  320          1167  2 !-
;  321          1168  2     BAS$$CB_POP ();
;  322          1169  2 !+
;  323          1170  2 ! Pop the "foreign buffer" CB if necessary.  It is kept on the CB stack until
;  324          1171  2 ! now to guard against an AST closing the foreign buffer channel.
;  325          1172  2 !-
;  326          1173  2
;  327          1174  3     IF (.TEMP_R11 NEQA 0)
;  328          1175  2     THEN
```

```
;  329          1176  3          BEGIN
;  330          1177  3          CCB = .TEMP_R11;
;  331          1178  3          BAS$$CB_POP ();
;  332          1179  2          END;
;  333          1180  2
;  334          1181  1      END;                                    !End of BAS$GET


                                                                    .TITLE  BAS$GET
                                                                    .IDENT  \1-021\

                                                                    .EXTRN  BAS$$OPEN_ZERO, BAS$K_IO_CHANOT
                                                                    .EXTRN  BAS$K_ILLILLACC
                                                                    .EXTRN  BAS$K_ILLIO_CHA
                                                                    .EXTRN  BAS$K_ILLRECLOC
                                                                    .EXTRN  BAS$$REC_GIN, BAS$$REC_GRE
                                                                    .EXTRN  BAS$$REC_GSE, BAS$$REC_GRFA
                                                                    .EXTRN  BAS$$CB_PUSH, BAS$$CB_POP
                                                                    .EXTRN  BAS$$STOP_IO, BAS$$STOP

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

                                                0BFC 00000          .ENTRY  BAS$GET, Save R2,R3,R4,R5,R6,R7,R8,R9,R11   ; 1005
                59 00000000G   00    9E 00002          MOVAB   BAS$$CB_POP, R9                                 ;
                58 00000000G   00    9E 00009          MOVAB   BAS$$CB_PUSH, R8                                ;
                57 00000000G   00    9E 00010          MOVAB   BAS$$STOP, R7                                   ;
                56 00000000G   00    9E 00017          MOVAB   BAS$$STOP_IO, R6                                ;
                5B       04    AC    D0 0001E          MOVL    UNIT, R11                                       ; 1069
                               07    18 00022          BGEQ    1$                                              ;
                7E       00G   8F    9A 00024          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                         ;
                               67    01 FB 00028          CALLS   #1, BAS$$STOP                                   ;
                               53          5D D0 0002B 1$:     MOVL    FP, FMP                                 ; 1071
                               55          D4 0002E            CLRL    TEMP_R11                                ; 1081
                               54          5B D0 00030         MOVL    R11, ACTUAL_UNIT                        ; 1082
                    00000077   8F          5B D1 00033         CMPL    R11, #119                               ; 1084
                               3E          15 0003A            BLEQ    4$                                      ;
             52                5B 00000100   8F C7 0003C      DIVL3   #256, R11, FOREIGN_BUFFER               ; 1091
          7E 00                5B          01 7A 00044         EMUL    #1, R11, #0, -(SP)                      ; 1092
       54 54                   8E 00000100   8F 7B 00049      EDIV    #256, (SP)+, ACTUAL_UNIT, ACTUAL_UNIT   ;
                    0000007F   8F          52 D1 00052         CMPL    FOREIGN_BUFFER, #127                    ; 1094
                               07          1B 00059            BLEQU   2$                                      ;
                7E       00G   8F    9A 0005B          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                         ;
                               67          01 FB 0005F         CALLS   #1, BAS$$STOP                           ;
                               50          D4 00062 2$:        CLRL    R0                                      ; 1096
                               68          16 00064            JSB     BAS$$CB_PUSH                            ;
                      FF4C     CB    0C    A3 D0 00066       MOVL    12(FMP), -180(CCB)                      ; 1097
                               07    FC    AB E8 0006C       BLBS    -4(CCB), 3$                             ; 1099
                7E       00G   8F    9A 00070          MOVZBL  #BAS$K_IO_CHANOT, -(SP)                         ;
                               66          01 FB 00074         CALLS   #1, BAS$$STOP_IO                        ;
                               55          5B D0 00077 3$:     MOVL    CCB, TEMP_R11                           ; 1101
                               54          D5 0007A 4$:        TSTL    ACTUAL_UNIT                             ; 1109
                               03          12 0007C            BNEQ    5$                                      ;
                               54          07 CE 0007E         MNEGL   #7, ACTUAL_UNIT                         ;
                               50          08 CE 00081 5$:     MNEGL   #8, R0                                  ; 1111
                               52          54 D0 00084         MOVL    ACTUAL_UNIT, R2                         ;
                                           68 16 00087         JSB     BAS$$CB_PUSH                            ;
                      FF4C     CB    0C    A3 D0 00089       MOVL    12(FMP), -180(CCB)                      ; 1112
```

```
                                17      FC    AB  E8 0008F          BLBS    -4(CCB), 7$                             ; 1119
                                54      D5 00093                   TSTL    ACTUAL_UNIT                             ; 1122
                                0C      18 00095                   BGEQ    6$
                                0C      A3  DD 00097               PUSHL   12(FMP)                                 ; 1125
                    00000000G   00          01  FB 0009A           CALLS   #1, BAS$$OPEN_ZERO
                                07          11 000A1               BRB     7$                                      ; 1124
                                7E      0G  8F  9A 000A3 6$:       MOVZBL  #BAS$K_IO_CHANOT, -(SP)                 ; 1129
                                66          01  FB 000A7           CALLS   #1, BAS$$STOP_IO
                            FF71    CB          24  90 000AA 7$:    MOVB    #36, -143(CCB)                          ; 1136
                                07      FF  AB  E9 000AF           BLBC    -1(CCB), 8$                             ; 1140
                                7E      00G 8F  9A 000B3           MOVZBL  #BAS$K_ILLILLACC, -(SP)
                                66          01  FB 000B7           CALLS   #1, BAS$$STOP_IO
                                FF      AB          02  88 000BA 8$:    BISB2   #2, -1(CCB)                         ; 1141
                                02          6C  91 000BE           CMPB    (AP), #2                                ; 1143
                                            04  1E 000C1           BGEQU   9$
                                            52  D4 000C3           CLRL    FLAGS                                   ; 1145
                                            20  11 000C5           BRB     13$
            53      06  AB                  02  EF 000C7 9$:       EXTZV   #2, #1, 6(CCB), R3                      ; 1151
                        01              00      53  CF 000CD       CASEL   R3, #0, #1
                                    0012            0004    000D1 10$:  .WORD   11$-10$,-
                                                                            12$-10$
                        09      08  AC              03  E0 000D5 11$:  BBS     #3, LOCK_FLAGS, 12$                 ; 1154
                                7E      00G 8F  9A 000DA           MOVZBL  #BAS$K_ILLRECLOC, -(SP)                 ; 1158
                                66          01  FB 000DE           CALLS   #1, BAS$$STOP_IO
                                            04  11 000E1           BRB     13$                                     ; 1154
                                52      08  AC  D0 000E3 12$:      MOVL    LOCK_FLAGS, FLAGS                       ; 1161
                                51          52  D0 000E7 13$:      MOVL    FLAGS, R1                               ; 1164
                                50          55  D0 000EA           MOVL    TEMP_R11, R0
                                    00000000G   00  16 000ED       JSB     BAS$$REC_GSE
                                            69  16 000F3           JSB     BAS$$CB_POP                             ; 1168
                                            55  D5 000F5           TSTL    TEMP_R11                                ; 1174
                                            05  13 000F7           BEQL    14$                                     ;
                                5B          55  D0 000F9           MOVL    TEMP_R11, CCB                           ; 1177
                                            69  16 000FC           JSB     BAS$$CB_POP                             ; 1178
                                            04 000FE 14$:          RET                                             ; 1181

; Routine Size:  255 bytes,    Routine Base:  _BAS$CODE + 0000

;  335          1182  1
```

```
;  337          1183  1 GLOBAL ROUTINE BAS$GET_KEY (                  ! GET indexed
;  338          1184  1         UNIT,                                 ! logical unit number
;  339          1185  1         KEY_NO,                               ! key of index
;  340          1186  1         REL_OP,                               ! relative relationship of keys
;  341          1187  1         KEY,                                  ! key to compare for
;  342          1188  1         LOCK_FLAGS                            ! manual locking bits
;  343          1189  1     ) : NOVALUE =
;  344          1190  1
;  345          1191  1 !++
;  346          1192  1 ! FUNCTIONAL DESCRIPTION:
;  347          1193  1 !
;  348          1194  1 !       This routine will set up the I/O data base for this LUN if necessary
;  349          1195  1 !       and then go directly to the REC level.  When control is returned to
;  350          1196  1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
;  351          1197  1 !       face to RMS is done at the REC level.  One record is read.
;  352          1198  1 !
;  353          1199  1 ! FORMAL PARAMETERS:
;  354          1200  1 !
;  355          1201  1 !       UNIT.rlu.v      logical unit number
;  356          1202  1 !       KEY_NO.rl.v
;  357          1203  1 !       REL_OP.rl.v
;  358          1204  1 !       KEY.rt.dx
;  359          1205  1 !       [LOCK_FLAGS.rlu.v]  if present, bits to pass on to record level to
;  360          1206  1 !                           control manual record locking
;  361          1207  1 !
;  362          1208  1 ! IMPLICIT INPUTS:
;  363          1209  1 !
;  364          1210  1 !       LUB$V_VA_USE                    virtual array use of this file
;  365          1211  1 !
;  366          1212  1 ! IMPLICIT OUTPUTS:
;  367          1213  1 !
;  368          1214  1 !       LUB$V_BLK_USE           non-virtual use of this file
;  369          1215  1 !       OTS$$A_CUR_LUB          pointer to current control block
;  370          1216  1 !       RECOUNT                 Basic Global which contains the number of bytes read
;  371          1217  1 !       ISB$B_STTM_TYPE         the statement type
;  372          1218  1 !
;  373          1219  1 ! COMPLETION CODES:
;  374          1220  1 !
;  375          1221  1 !       NONE
;  376          1222  1 !
;  377          1223  1 ! SIDE EFFECTS:
;  378          1224  1 !
;  379          1225  1 !       RECOUNT is assigned the number of bytes read.
;  380          1226  1 !       Signals:
;  381          1227  1 !       BAS$K_IO_CHANOT (I/O channel not open)
;  382          1228  1 !       BAS$K_ILLIO_CHA (illegal I/O channel) for foreign buffers.
;  383          1229  1 !       BAS$K_ILLILLACC (illegal or illogical access)
;  384          1230  1 !
;  385          1231  1 !--
;  386          1232  1
;  387          1233  2     BEGIN
;  388          1234  2
;  389          1235  2     BUILTIN
;  390          1236  2         FP,
;  391          1237  2         ACTUALCOUNT;
;  392          1238  2
;  393          1239  2     LITERAL
```

```
;  394          1240  2          K_LOCK_ARG = 5;
;  395          1241  2
;  396          1242  2      GLOBAL REGISTER
;  397          1243  2          CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  398          1244  2
;  399          1245  2      LOCAL
;  400          1246  2          FMP : REF BLOCK [, BYTE],
;  401          1247  2          ACTUAL_UNIT,                        ! Unit number, without foreign buffer
;  402          1248  2          TEMP_R11,                           ! CCB for foreign buffer, or 0
;  403          1249  2          FLAGS;
;  404          1250  2
;  405          1251  2      FMP = .FP;
;  406          1252  2  !+
;  407          1253  2  ! Check for "foreign buffers".  If the unit number exceeds 255 then a foreign
;  408          1254  2  ! buffer is specified.  The foreign buffer is actually a unit number whose
;  409          1255  2  ! buffer is to receive the record which is read.  The "foreign buffer" unit
;  410          1256  2  ! is pushed to pick up the CB address which is passed to the REC level.  Then
;  411          1257  2  ! the unit pointing to the file is pushed so that the CCB points to the log-
;  412          1258  2  ! ical unit which actually do the I/O.  Upon return, the necessary RAB fields
;  413          1259  2  ! (USZ and UBF) have been restored and two CB_POPs are done if necessary.
;  414          1260  2  !-
;  415          1261  2      TEMP_R11 = 0;
;  416          1262  2      ACTUAL_UNIT = .UNIT;
;  417          1263  2
;  418          1264  3      IF (.UNIT GTR LUB$K_LUN_MAX)
;  419          1265  2      THEN
;  420          1266  3          BEGIN
;  421          1267  3
;  422          1268  3          LOCAL
;  423          1269  3              FOREIGN_BUFFER;
;  424          1270  3
;  425          1271  3          FOREIGN_BUFFER = .UNIT/BAS$K_LUN_MAX;
;  426          1272  3          ACTUAL_UNIT = .UNIT MOD BAS$K_LUN_MAX;
;  427          1273  3
;  428          1274  3          IF (.FOREIGN_BUFFER GTRU BAS$K_MAX_FOR_B) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
;  429          1275  3
;  430          1276  3          BAS$$CB_PUSH (.FOREIGN_BUFFER, LUB$K_LUN_MIN);
;  431          1277  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  432          1278  3
;  433          1279  3          IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  434          1280  3
;  435          1281  3          TEMP_R11 = .CCB;
;  436          1282  2          END;
;  437          1283  2
;  438          1284  2      BAS$$CB_PUSH (.ACTUAL_UNIT, LUB$K_ILUN_MIN);
;  439          1285  2      CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  440          1286  2  !+
;  441          1287  2  ! If the channel is not open, give an error.
;  442          1288  2  !-
;  443          1289  2
;  444          1290  2      IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  445          1291  2
;  446          1292  2  !+
;  447          1293  2  ! Now that the data base is in place, store the statement type and go
;  448          1294  2  ! directly to the REC level.
;  449          1295  2  !-
;  450          1296  2      CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_GIN;
```

```
;  451          1297  2 !+
;  452          1298  2 ! Check for virtual array usage and set block usage.
;  453          1299  2 !-
;  454          1300  2     IF .CCB [LUB$V_VA_USE] EQL 1 THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  455          1301  2     CCB [LUB$V_BLK_USE] = 1;
;  456          1302  2
;  457          1303  2     IF ACTUALCOUNT () LSS K_LOCK_ARG
;  458          1304  2     THEN
;  459          1305  2         FLAGS = 0
;  460          1306  2     ELSE
;  461          1307  3         BEGIN
;  462          1308  3         !+
;  463          1309  3         ! The ULK bit must set unless this is a REGARDLESS clause.
;  464          1310  3         !-
;  465          1311  3         CASE .CCB [RAB$V_ULK] FROM 0 TO 1 OF
;  466          1312  3         SET
;  467          1313  3             [0]:
;  468          1314  3             IF (.LOCK_FLAGS AND RAB$M_RRL) NEQ 0
;  469          1315  3             THEN
;  470          1316  3                 FLAGS = .LOCK_FLAGS
;  471          1317  3             ELSE
;  472          1318  3                 BAS$$STOP_IO (BAS$K_ILLRECLOC);
;  473          1319  3
;  474          1320  3             [1]:
;  475          1321  3             FLAGS = .LOCK_FLAGS;
;  476          1322  3         TES;
;  477          1323  2         END;
;  478          1324  2     BAS$$REC_GIN (.KEY_NO, .REL_OP, .KEY, .TEMP_R11, .FLAGS);
;  479          1325  2 !+
;  480          1326  2 ! Now that the GET has been done, pop the CCB off the I/O system.
;  481          1327  2 !-
;  482          1328  2     BAS$$CB_POP ();
;  483          1329  2 !+
;  484          1330  2 ! Pop the "foreign buffer" CB if necessary.  It is kept on the CB stack until
;  485          1331  2 ! now to guard against an AST closing the foreign buffer channel.
;  486          1332  2 !-
;  487          1333  2
;  488          1334  3     IF (.TEMP_R11 NEQA 0)
;  489          1335  2     THEN
;  490          1336  3         BEGIN
;  491          1337  3         CCB = .TEMP_R11;
;  492          1338  3         BAS$$CB_POP ();
;  493          1339  2         END;
;  494          1340  2
;  495          1341  1     END;                                    !End of BAS$GET_KEY


                                    09FC 00000          .ENTRY  BAS$GET_KEY, Save R2,R3,R4,R5,R6,R7,R8,R11  ; 1183
58 00000000G    00    9E 00002      MOVAB   BAS$$CB_POP, R8                                                 ;
57 00000000G    00    9E 00009      MOVAB   BAS$$CB_PUSH, R7                                                ;
56 00000000G    00    9E 00010      MOVAB   BAS$$STOP_IO, R6                                                ;
54              5D    D0 00017      MOVL    FP, FMP                                                         ; 1251
                53    D4 0001A      CLRL    TEMP_R11                                                        ; 1261
55        04    AC    D0 0001C      MOVL    UNIT, ACTUAL_UNIT                                               ; 1262
```

```
                00000077   8F        04   AC  D1 00020          CMPL    UNIT, #119                                          ; 1264
                                           44  15 00028          BLEQ    3$
                52         04   AC 00000100  8F  C7 0002A      3$: DIVL3   #256, UNIT, FOREIGN_BUFFER                          ; 1271
             7E 00         04   AC           01  7A 00033          EMUL    #1, UNIT, #0, -(SP)                                 ; 1272
             55 55              8E 00000100  8F  7B 00039          EDIV    #256, (SP)+, ACTUAL_UNIT, ACTUAL_UNIT
                0000007F   8F                52  D1 00042          CMPL    FOREIGN_BUFFER, #127                                ; 1274
                                           0B  1B 00049          BLEQU   1$
                           7E        00G   8F  9A 0004B          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)
                00000000G  00              01  FB 0004F          CALLS   #1, BAS$$STOP
                                           50  D4 00056      1$: CLRL    R0                                                  ; 1276
                                           67  16 00058          JSB     BAS$$CB_PUSH
                     FF4C  CB        0C    A4  D0 0005A          MOVL    12(FMP), -180(CCB)                                  ; 1277
                           07        FC    AB  E8 00060          BLBS    -4(CCB), 2$                                         ; 1279
                           7E        00G   8F  9A 00064          MOVZBL  #BAS$K_IO_CHANOT, -(SP)
                           66              01  FB 00068          CALLS   #1, BAS$$STOP_IO
                           53              5B  D0 0006B      2$: MOVL    CCB, TEMP_R11                                       ; 1281
                           50              08  CE 0006E      3$: MNEGL   #8, R0                                              ; 1284
                           52              55  D0 00071          MOVL    ACTUAL_UNIT, R2
                                           67  16 00074          JSB     BAS$$CB_PUSH
                     FF4C  CB        0C    A4  D0 00076          MOVL    12(FMP), -180(CCB)                                  ; 1285
                           07        FC    AB  E8 0007C          BLBS    -4(CCB), 4$                                         ; 1290
                           7E        00G   8F  9A 00080          MOVZBL  #BAS$K_IO_CHANOT, -(SP)
                           66              01  FB 00084          CALLS   #1, BAS$$STOP_IO
                     FF71  CB              2C  90 00087      4$: MOVB    #44, -143(CCB)                                      ; 1296
                           07        FF    AB  E9 0008C          BLBC    -1(CCB), 5$                                         ; 1300
                           7E        00G   8F  9A 00090          MOVZBL  #BAS$K_ILLILLACC, -(SP)
                           66              01  FB 00094          CALLS   #1, BAS$$STOP_IO
                        FF   AB            02  88 00097      5$: BISB2   #2, -1(CCB)                                         ; 1301
                           05              6C  91 0009B          CMPB    (AP), #5                                            ; 1303
                                           04  1E 0009E          BGEQU   6$
                                           54  D4 000A0          CLRL    FLAGS                                               ; 1305
                                           20  11 000A2          BRB     10$
             52         06   AB              01  EF 000A4      6$: EXTZV   #2, #1, 6(CCB), R2                                  ; 1311
                        01                 00  CF 000AA          CASEL   R2, #0, #1
                                      0012      0004  000AE      7$: .WORD   8$-7$,-
                                                                       9$-7$
                           09   14   AC          03  E0 000B2      8$: BBS     #3, LOCK_FLAGS, 9$                              ; 1314
                                     7E    00G   8F  9A 000B7          MOVZBL  #BAS$K_ILLRECLOC, -(SP)                         ; 1318
                                     66          01  FB 000BB          CALLS   #1, BAS$$STOP_IO
                                                 04  11 000BE          BRB     10$                                             ; 1314
                                     54    14    AC  D0 000C0      9$: MOVL    LOCK_FLAGS, FLAGS                               ; 1321
                                     51    0C    AC  7D 000C4     10$: MOVQ    REL_OP, R1                                      ; 1324
                                     50    08    AC  D0 000C8          MOVL    KEY_NO, R0
                                        00000000G  00  16 000CC          JSB     BAS$$REC_GIN
                                                   68  16 000D2          JSB     BAS$$CB_POP                                 ; 1328
                                                   53  D5 000D4          TSTL    TEMP_R11                                    ; 1334
                                                   05  13 000D6          BEQL    11$
                                     5B            53  D0 000D8          MOVL    TEMP_R11, CCB                               ; 1337
                                                   68  16 000DB          JSB     BAS$$CB_POP                                 ; 1338
                                                       04 000DD     11$: RET                                                 ; 1341

; Routine Size:  222 bytes,    Routine Base:  _BAS$CODE + 00FF

;  496          1342  1
```

```
;  498          1343  1 GLOBAL ROUTINE BAS$GET_RECORD (                       ! GET relative
;  499          1344  1         UNIT,                                         ! logical unit number
;  500          1345  1         RECORD_NUM,                                   ! relative record number
;  501          1346  1         LOCK_FLAGS                                    ! manual locking bits
;  502          1347  1     ) : NOVALUE =
;  503          1348  1
;  504          1349  1 !++
;  505          1350  1 ! FUNCTIONAL DESCRIPTION:
;  506          1351  1 !
;  507          1352  1 !       This routine will set up the I/O data base for this LUN if necessary
;  508          1353  1 !       and then dispatch off to the UDF level.  When control is returned to
;  509          1354  1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
;  510          1355  1 !       face to RMS is done at the REC level.  One record is read.
;  511          1356  1 !       NOTE: Foreign buffers apply to GET and PUT.  The LUN of the foreign buffer
;  512          1357  1 !       is in the upper byte of the unit number.
;  513          1358  1 !
;  514          1359  1 ! FORMAL PARAMETERS:
;  515          1360  1 !
;  516          1361  1 !       UNIT.rlu.v      logical unit number
;  517          1362  1 !       RECORD_NUM.rl.v         relative record number
;  518          1363  1 !       [LOCK_FLAGS.rlu.v]      if present, bits to pass on to record level to
;  519          1364  1 !                               control manual record locking
;  520          1365  1 !
;  521          1366  1 ! IMPLICIT INPUTS:
;  522          1367  1 !
;  523          1368  1 !       OTS$$A_CUR_LUB          pointer to current control block
;  524          1369  1 !       LUB$V_VA_USE            indicates virtual array usage
;  525          1370  1 !
;  526          1371  1 ! IMPLICIT OUTPUTS:
;  527          1372  1 !
;  528          1373  1 !       LUB$V_BLK_USE           indicates non-virtual array usage
;  529          1374  1 !       RECOUNT                 Basic Global which contains the number of bytes read
;  530          1375  1 !       ISB$B_STTM_TYPE         the statement type
;  531          1376  1 !
;  532          1377  1 ! COMPLETION CODES:
;  533          1378  1 !
;  534          1379  1 !       NONE
;  535          1380  1 !
;  536          1381  1 ! SIDE EFFECTS:
;  537          1382  1 !
;  538          1383  1 !       RECOUNT is assigned the number of bytes read.
;  539          1384  1 !       Signals:
;  540          1385  1 !       BAS$K_IO_CHANOT (I/O channel no open)
;  541          1386  1 !       BAS$K_ILLIO_CHA (Illegal I/O channel)
;  542          1387  1 !       for foreign buffers
;  543          1388  1 !       BAS$K_ILLILLACC (illegal or illogical access)
;  544          1389  1 !
;  545          1390  1 !--
;  546          1391  1
;  547          1392  2     BEGIN
;  548          1393  2
;  549          1394  2     BUILTIN
;  550          1395  2         FP,
;  551          1396  2         ACTUALCOUNT;
;  552          1397  2
;  553          1398  2     GLOBAL REGISTER
;  554          1399  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
```

```
;  555          1400  2
;  556          1401  2      LOCAL
;  557          1402  2          FMP : REF BLOCK [, BYTE],
;  558          1403  2          ACTUAL_UNIT,                    ! Unit number, without foreign buffer
;  559          1404  2          TEMP_R11,                       ! CCB for foreign buffer, or 0
;  560          1405  2          FLAGS;
;  561          1406  2
;  562          1407  2      LITERAL
;  563          1408  2          K_LOCK_ARG = 3;
;  564          1409  2
;  565          1410  2      FMP = .FP;
;  566          1411  2  !+
;  567          1412  2  ! Check for "foreign buffers".  If the unit number exceeds 255 then a foreign
;  568          1413  2  ! buffer is specified.  The foreign buffer is actually a unit number whose
;  569          1414  2  ! buffer is to receive the record which is read.  The "foreign buffer" unit
;  570          1415  2  ! is pushed to pick up the CB address which is passed to the REC level.  Then
;  571          1416  2  ! the unit pointing to the file is pushed so that the CCB points to the log-
;  572          1417  2  ! ical unit which actually do the I/O.  Upon return, the necessary RAB fields
;  573          1 18  2  ! (USZ and UBF) have been restored and two CB_POPs are done if necessary.
;  574          1419  2  !-
;  575          1420  2      TEMP_R11 = 0;
;  576          1421  2      ACTUAL_UNIT = .UNIT;
;  577          1422  2
;  578          1423  3      IF (.UNIT GTR LUB$K_LUN_MAX)
;  579          1424  2      THEN
;  580          1425  3          BEGIN
;  581          1426  3
;  582          1427  3          LOCAL
;  583          1428  3              FOREIGN_BUFFER;
;  584          1429  3
;  585          1430  3          FOREIGN_BUFFER = .UNIT/BAS$K_LUN_MAX;
;  586          1431  3          ACTUAL_UNIT = .UNIT MOD BAS$K_LUN_MAX;
;  587          1432  3
;  588          1433  3          IF (.FOREIGN_BUFFER GTRU BAS$K_MAX_FOR_B) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
;  589          1434  3
;  590          1435  3          BAS$$CB_PUSH (.FOREIGN_BUFFER, LUB$K_LUN_MIN);
;  591          1436  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  592          1437  3
;  593          1438  3          IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  594          1439  3
;  595          1440  3          TEMP_R11 = .CCB;
;  596          1441  2          END;
;  597          1442  2
;  598          1443  2      BAS$$CB_PUSH (.ACTUAL_UNIT, LUB$K_ILUN_MIN);
;  599          1444  2      CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  600          1445  2  !+
;  601          1446  2  ! If the channel is not open, give an error.
;  602          1447  2  !-
;  603          1448  2
;  604          1449  2      IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  605          1450  2
;  606          1451  2  !+
;  607          1452  2  ! Now that the data base is in place, store the statement type, store the index, and  go
;  608          1453  2  ! directly to the REC level.
;  609          1454  2  !-
;  610          1455  2      CCB [LUB$L_LOG_RECNO] = .RECORD_NUM;
;  611          1456  2      CCB [ISB$B_STM_TYPE] = ISB$K_ST_TY_GRE;
```

```
;  612          1457  2 !+
;  613          1458  2 ! Check for virtual array usage and set block usage.
;  614          1459  2 !-
;  615          1460  2     IF .CCB [LUB$V_VA_USE] EQL 1 THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  616          1461  2     CCB [LUB$V_BLK_USE] = 1;
;  617          1462  2
;  618          1463  2     IF ACTUALCOUNT () LSS K_LOCK_ARG
;  619          1464  2     THEN
;  620          1465  2         FLAGS = 0
;  621          1466  2     ELSE
;  622          1467  3         BEGIN
;  623          1468  3         !+
;  624          1469  3         ! The ULK bit must set unless this is a REGARDLESS clause.
;  625          1470  3         !-
;  626          1471  3         CASE .CCB [RAB$V_ULK] FROM 0 TO 1 OF
;  627          1472  3         SET
;  628          1473  3             [0]:
;  629          1474  3             IF (.LOCK_FLAGS AND RAB$M_RRL) NEQ 0
;  630          1475  3             THEN
;  631          1476  3                 FLAGS = .LOCK_FLAGS
;  632          1477  3             ELSE
;  633          1478  3                 BAS$$STOP_IO (BAS$K_ILLRECLOC);
;  634          1479  3
;  635          1480  3             [1]:
;  636          1481  3             FLAGS = .LOCK_FLAGS;
;  637          1482  3         TES;
;  638          1483  3         END;
;  639          1484  2     BAS$$REC_GRE (.TEMP_R11, .FLAGS);
;  640          1485  2 !+
;  641          1486  2 ! Now that the GET has been done, pop the CCB off the I/O system.
;  642          1487  2 !-
;  643          1488  2     BAS$$CB_POP ();
;  644          1489  2 !+
;  645          1490  2 ! Pop the "foreign buffer" CB if necessary.  It is kept on the CB stack until
;  646          1491  2 ! now to guard against an AST closing the foreign buffer channel.
;  647          1492  2 !-
;  648          1493  2
;  649          1494  3     IF (.TEMP_R11 NEQA 0)
;  650          1495  2     THEN
;  651          1496  3         BEGIN
;  652          1497  3         CCB = .TEMP_R11;
;  653          1498  3         BAS$$CB_POP ();
;  654          1499  2         END;
;  655          1500  2
;  656          1501  1     END;                                    !End of BAS$GET_RECORD


                                    09FC 00000          .ENTRY  BAS$GET_RECORD, Save R2,R3,R4,R5,R6,R7,R8,- ; 1343
                                                                R11                                         ;
                58 00000000G   00   9E 00002            MOVAB   BAS$$CB_POP, R8                             ;
                57 00000000G   00   9E 00009            MOVAB   BAS$$CB_PUSH, R7                            ;
                56 00000000G   00   9E 00010            MOVAB   BAS$$STOP_IO, R6                            ;
                53             5D   D0 00017            MOVL    FP, FMP                                     ; 1410
                               55   D4 0001A            CLRL    TEMP_R11                                    ; 1420
```

```
                        54        04   AC  D0 0001C          MOVL    UNIT, ACTUAL_UNIT                                   ; 1421
                00000077  8F      04   AC  D1 00020          CMPL    UNIT, #119                                          ; 1423
                                       44  15 00028          BLEQ    3$
                     52   04   AC 00000100  8F  C7 0002A     DIVL3   #256, UNIT, FOREIGN_BUFFER                          ; 1430
                7E   00   04   AC          01  7A 00033     EMUL    #1, UNIT, #0, -(SP)                                 ; 1431
                54   54                    8E  7B 00039     EDIV    #256, (SP)+, ACTUAL_UNIT, ACTUAL_UNIT
                0000007F  8F               52  D1 00042     CMPL    FOREIGN_BUFFER, #127                                ; 1433
                                           0B  1B 00049     BLEQU   1$
                        7E        00G      8F  9A 0004B     MOVZBL  #BAS$K_ILLIO_CHA, -(SP)
                00000000G 00               01  FB 0004F     CALLS   #1, BAS$$STOP
                                           50  D4 00056 1$:  CLRL    R0                                                  ; 1435
                                           67  16 00058     JSB     BAS$$CB_PUSH
                    FF4C  CB       0C      A3  D0 0005A     MOVL    12(FMP), -180(CCB)                                  ; 1436
                          07       FC      AB  E8 0C060     BLBS    -4(CCB), 2$                                         ; 1438
                          7E       00G     8F  9A 00064     MOVZBL  #BAS$K_IO_CHANOT, -(SP)
                          66               01  FB 00068     CALLS   #1, BAS$$STOP_IO
                          55               5B  D0 0006B 2$:  MOVL    CCB, TEMP_R11                                       ; 1440
                          50               08  CE 0006E 3$:  MNEGL   #8, R0                                              ; 1443
                          52               54  D0 00071     MOVL    ACTUAL_UNIT, R2
                                           67  16 00074     JSB     BAS$$CB_PUSH
                    FF4C  CB       0C      A3  D0 00076     MOVL    12(FMP), -180(CCB)                                  ; 1444
                          07       FC      AB  E8 0007C     BLBS    -4(CCB), 4$                                         ; 1449
                          7E       00G     8F  9A 00080     MOVZBL  #BAS$K_IO_CHANOT, -(SP)
                          66               01  FB 00084     CALLS   #1, BAS$$STOP_IO
                      E0  AB       08      AC  D0 00087 4$:  MOVL    RECORD_NUM, -32(CCB)                                ; 1455
                    FF71  CB               28  90 0008C     MOVB    #40, -143(CCB)                                      ; 1456
                          07       FF      AB  E9 00091     BLBC    -1(CCB), 5$                                         ; 1460
                          7E       00G     8F  9A 00095     MOVZBL  #BAS$K_ILLILLACC, -(SP)
                          66               01  FB 00099     CALLS   #1, BAS$$STOP_IO
                       FF  AB              02  88 0009C 5$:  BISB2   #2, -1(CCB)                                         ; 1461
                          03               6C  91 000A0     CMPB    (AP), #3                                            ; 1463
                                           04  1E 000A3     BGEQU   6$
                                           52  D4 000A5     CLRL    FLAGS                                               ; 1465
                                           20  11 000A7     BRB     10$
                53        06   AB              01  EF 000A9 6$:  EXTZV   #2, #1, 6(CCB), R3                            ; 1471
                          01               00  53  CF 000AF     CASEL   R3, #0, #1
                                     0012      0004     000B3 7$:  .WORD   8$-7$,-
                                                                   9$-7$
                          09       0C  AC          03  E0 000B7 8$:  BBS     #3, LOCK_FLAGS, 9$                          ; 1474
                                       7E      00G 8F  9A 000BC     MOVZBL  #BAS$K_ILLRECLOC, -(SP)                     ; 1478
                                       66          01  FB 000C0     CALLS   #1, BAS$$STOP_IO
                                                   04  11 000C3     BRB     10$                                         ; 1474
                                       52      0C  AC  D0 000C5 9$:  MOVL    LOCK_FLAGS, FLAGS                           ; 1481
                                       51          52  D0 000C9 10$: MOVL    FLAGS, R1                                   ; 1484
                                       50          55  D0 000CC     MOVL    TEMP_R11, R0
                                            00000000G  00  16 000CF     JSB     BAS$$REC_GRE
                                                       68  16 000D5     JSB     BAS$$CB_POP                             ; 1488
                                                       55  D5 000D7     TSTL    TEMP_R11                                ; 1494
                                                       05  13 000D9     BEQL    11$
                                       5B              55  D0 000DB     MOVL    TEMP_R11, CCB                           ; 1497
                                                       68  16 000DE     JSB     BAS$$CB_POP                             ; 1498
                                                           04 000E0 11$: RET                                             ; 1501

; Routine Size:  225 bytes,    Routine Base:  _BAS$CODE + 01DD
```

```
;  658          1502  1 GLOBAL ROUTINE BAS$GET_RFA (                  ! GET by RFA
;  659          1503  1         UNIT,                                 ! logical unit number
;  660          1504  1         RFA,                                  ! RFA
;  661          1505  1         LOCK_FLAGS                            ! manual locking bits
;  662          1506  1     ) : NOVALUE =
;  663          1507  1
;  664          1508  1 !++
;  665          1509  1 ! FUNCTIONAL DESCRIPTION:
;  666          1510  1 !
;  667          1511  1 !       This routine will set up the I/O data base for this LUN if necessary
;  668          1512  1 !       and then dispatch off to the UDF level.  When control is returned to
;  669          1513  1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
;  670          1514  1 !       face to RMS is done at the REC level.  One record is read.
;  671          1515  1 !       NOTE: Foreign buffers apply to GET and PUT.  The LUN of the foreign buffer
;  672          1516  1 !       is in the upper byte of the unit number.
;  673          1517  1 !
;  674          1518  1 ! FORMAL PARAMETERS:
;  675          1519  1 !
;  676          1520  1 !       UNIT.rlu.v      logical unit number
;  677          1521  1 !       RFA_DESC.rx.r           RFA address
;  678          1522  1 !       [LOCK_FLAGS.rlu.v]      if present, bits to pass on to record level to
;  679          1523  1 !                               control manual record locking
;  680          1524  1 !
;  681          1525  1 ! IMPLICIT INPUTS:
;  682          1526  1 !
;  683          1527  1 !       OTS$$A_CUR_LUB          pointer to current control block
;  684          1528  1 !       LUB$V_VA_USE            indicates virtual array usage
;  685          1529  1 !
;  686          1530  1 ! IMPLICIT OUTPUTS:
;  687          1531  1 !
;  688          1532  1 !       LUB$V_BLK_USE           indicates non-virtual array usage
;  689          1533  1 !       RECOUNT                 Basic Global which contains the number of bytes read
;  690          1534  1 !       ISB$B_STTM_TYPE         the statement type
;  691          1535  1 !
;  692          1536  1 ! COMPLETION CODES:
;  693          1537  1 !
;  694          1538  1 !       NONE
;  695          1539  1 !
;  696          1540  1 ! SIDE EFFECTS:
;  697          1541  1 !
;  698          1542  1 !       RECOUNT is assigned the number of bytes read.
;  699          1543  1 !       Signals:
;  700          1544  1 !       BAS$K_IO_CHANOT (I/O channel no open)
;  701          1545  1 !       BAS$K_ILLIO_CHA (Illegal I/O channel)
;  702          1546  1 !       for foreign buffers
;  703          1547  1 !       BAS$K_ILLILLACC (illegal or illogical access)
;  704          1548  1 !
;  705          1549  1 !--
;  706          1550  1
;  707          1551  2     BEGIN
;  708          1552  2
;  709          1553  2     BUILTIN
;  710          1554  2         FP,
;  711          1555  2         ACTUALCOUNT;
;  712          1556  2
;  713          1557  2     GLOBAL REGISTER
;  714          1558  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
```

```
;  715          1559  2
;  716          1560  2     LOCAL
;  717          1561  2         FMP : REF BLOCK [, BYTE],
;  718          1562  2         ACTUAL_UNIT,                        ! Unit number, without foreign buffer
;  719          1563  2         TEMP_R11,                           ! CCB for foreign buffer, or 0
;  720          1564  2         FLAGS;
;  721          1565  2
;  722          1566  2     LITERAL
;  723          1567  2         K_LOCK_ARG = 3;
;  724          1568  2
;  725          1569  2     FMP = .FP;
;  726          1570  2 !+
;  727          1571  2 ! Check for "foreign buffers".  If the unit number exceeds 255 then a foreign
;  728          1572  2 ! buffer is specified.  The foreign buffer is actually a unit number whose
;  729          1573  2 ! buffer is to receive the record which is read.  The "foreign buffer" unit
;  730          1574  2 ! is pushed to pick up the CB address which is passed to the REC level.  Then
;  731          1575  2 ! the unit pointing to the file is pushed so that the CCB points to the log-
;  732          1576  2 ! ical unit which actually do the I/O.  Upon return, the necessary RAB fields
;  733          1577  2 ! (USZ and UBF) have been restored and two CB_POPs are done if necessary.
;  734          1578  2 !-
;  735          1579  2     TEMP_R11 = 0;
;  736          1580  2     ACTUAL_UNIT = .UNIT;
;  737          1581  2
;  738          1582  3     IF (.UNIT GTR LUB$K_LUN_MAX)
;  739          1583  2     THEN
;  740          1584  3         BEGIN
;  741          1585  3
;  742          1586  3         LOCAL
;  743          1587  3             FOREIGN_BUFFER;
;  744          1588  3
;  745          1589  3         FOREIGN_BUFFER = .UNIT/BAS$K_LUN_MAX;
;  746          1590  3         ACTUAL_UNIT = .UNIT MOD BAS$K_LUN_MAX;
;  747          1591  3
;  748          1592  3         IF (.FOREIGN_BUFFER GTRU BAS$K_MAX_FOR_B) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
;  749          1593  3
;  750          1594  3         BAS$$CB_PUSH (.FOREIGN_BUFFER, LUB$K_LUN_MIN);
;  751          1595  3         CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  752          1596  3
;  753          1597  3         IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  754          1598  3
;  755          1599  3         TEMP_R11 = .CCB;
;  756          1600  2         END;
;  757          1601  2
;  758          1602  2     BAS$$CB_PUSH (.ACTUAL_UNIT, LUB$K_ILUN_MIN);
;  759          1603  2     CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  760          1604  2 !+
;  761          1605  2 ! If the channel is not open, give an error.
;  762          1606  2 !-
;  763          1607  2
;  764          1608  2     IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  765          1609  2
;  766          1610  2 !+
;  767          1611  2 ! Now that the data base is in place, store the statement type, store the index, and  go
;  768          1612  2 ! directly to the REC level.
;  769          1613  2 !-
;  770          1614  2     CH$MOVE (6, .RFA, CCB [RAB$W_RFA]);
;  771          1615  2     CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_GRFA;
```

```
;  772          1616  2 !+
;  773          1617  2 ! Check for virtual array usage and set block usage.
;  774          1618  2 !-
;  775          1619  2     IF .CCB [LUB$V_VA_USE] EQL 1 THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  776          1620  2     CCB [LUB$V_BLK_USE] = 1;
;  777          1621  2
;  778          1622  2     IF ACTUALCOUNT () LSS K_LOCK_ARG
;  779          1623  2     THEN
;  780          1624  2         FLAGS = 0
;  781          1625  2     ELSE
;  782          1626  3         BEGIN
;  783          1627  3         !+
;  784          1628  3         ! The ULK bit must set unless this is a REGARDLESS clause.
;  785          1629  3         !-
;  786          1630  3         CASE .CCB [RAB$V_ULK] FROM 0 TO 1 OF
;  787          1631  3         SET
;  788          1632  3             [0]:
;  789          1633  3             IF (.LOCK_FLAGS AND RAB$M_RRL) NEQ 0
;  790          1634  3             THEN
;  791          1635  3                 FLAGS = .LOCK_FLAGS
;  792          1636  3             ELSE
;  793          1637  3                 BAS$$STOP_IO (BAS$K_ILLRECLOC);
;  794          1638  3
;  795          1639  3             [1]:
;  796          1640  3             FLAGS = .LOCK_FLAGS;
;  797          1641  3         TES;
;  798          1642  2         END;
;  799          1643  2     BAS$$REC_GRFA (.TEMP_R11, .FLAGS);
;  800          1644  2 !+
;  801          1645  2 ! Now that the GET has been done, pop the CCB off the I/O system.
;  802          1646  2 !-
;  803          1647  2     BAS$$CB_POP ();
;  804          1648  2 !+
;  805          1649  2 ! Pop the "foreign buffer" CB if necessary.  It is kept on the CB stack until
;  806          1650  2 ! now to guard against an AST closing the foreign buffer channel.
;  807          1651  2 !-
;  808          1652  2
;  809          1653  3     IF (.TEMP_R11 NEQA 0)
;  810          1654  2     THEN
;  811          1655  3         BEGIN
;  812          1656  3         CCB = .TEMP_R11;
;  813          1657  3         BAS$$CB_POP ();
;  814          1658  2         END;
;  815          1659  2
;  816          1660  1     END;                                    !End of BAS$GET_RFA


                                    0BFC 00000          .ENTRY  BAS$GET_RFA, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 1502
                                                                R11                                          ;
                         59 00000000G   00   9E 00002           MOVAB   BAS$$CB_POP, R9                      ;
                         58 00000000G   00   9E 00009           MOVAB   BAS$$CB_PUSH, R8                     ;
                         57 00000000G   00   9E 00010           MOVAB   BAS$$STOP_IO, R7                     ;
                         53             5D   D0 00017           MOVL    FP, FMP                              ; 1569
                                        56   D4 0001A           CLRL    TEMP_R11                             ; 1579
```

```
                                54        04    AC  D0 0001C          MOVL    UNIT, ACTUAL_UNIT                                   ; 1580
                    00000077    8F        04    AC  D1 00020          CMPL    UNIT, #119                                          ; 1582
                                                  44  15 00028          BLEQ    3$                                                  ;
                          52        04    AC 00000100 8F  C7 0002A    DIVL3   #256, UNIT, FOREIGN_BUFFER                          ; 1589
                7E        00        04    AC          01  7A 00033    EMUL    #1, UNIT, #0, -(SP)                                 ; 1590
                54        54              8E 00000100 8F  7B 00039    EDIV    #256, (SP)+, ACTUAL_UNIT, ACTUAL_UNIT               ;
                    0000007F    8F                    52  D1 00042    CMPL    FOREIGN_BUFFER, #127                                ; 1592
                                                      0B  1B 00049    BLEQU   1$                                                  ;
                                7E              00G   8F  9A 0004B    MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                             ;
                    00000000G   00                    01  FB 0004F    CALLS   #1, BAS$$STOP                                       ;
                                                      50  D4 00056 1$: CLRL    R0                                                  ; 1594
                                                      68  16 00058    JSB     BAS$$CB_PUSH                                        ;
                        FF4C    CB        0C    A3  D0 0005A          MOVL    12(FMP), -180(CCB)                                  ; 1595
                                07        FC    AB  E8 00060          BLBS    -4(CCB), 2$                                         ; 1597
                                7E        00G   8F  9A 00064          MOVZBL  #BAS$K_IO_CHANOT, -(SP)                             ;
                                67              01  FB 00068          CALLS   #1, BAS$$STOP_IO                                    ;
                                56              5B  D0 0006B 2$:      MOVL    CCB, TEMP_R11                                       ; 1599
                                50              08  CE 0006E 3$:      MNEGL   #8, R0                                              ; 1602
                                52              54  D0 00071          MOVL    ACTUAL_UNIT, R2                                     ;
                                                68  16 00074          JSB     BAS$$CB_PUSH                                        ;
                        FF4C    CB        0C    A3  D0 00076          MOVL    12(FMP), -180(CCB)                                  ; 1603
                                07        FC    AB  E8 0007C          BLBS    -4(CCB), 4$                                         ; 1608
                                7E        00G   8F  9A 00080          MOVZBL  #BAS$K_IO_CHANOT, -(SP)                             ;
                                67              01  FB 00084          CALLS   #1, BAS$$STOP_IO                                    ;
                10  AB            08    BC          06  28 00087 4$:      MOVC3   #6, @RFA, 16(CCB)                                   ; 1614
                        FF71    CB              37  90 0008D          MOVB    #55, -143(CCB)                                      ; 1615
                                07        FF    AB  E9 00092          BLBC    -1(CCB), 5$                                         ; 1619
                                7E        00G   8F  9A 00096          MOVZBL  #BAS$K_ILLILLACC, -(SP)                             ;
                                67              01  FB 0009A          CALLS   #1, BAS$$STOP_IO                                    ;
                              FF    AB              02  88 0009D 5$:      BISB2   #2, -1(CCB)                                         ; 1620
                                03              6C  91 000A1          CMPB    (AP), #3                                            ; 1622
                                                04  1E 000A4          BGEQU   6$                                                  ;
                                                52  D4 000A6          CLRL    FLAGS                                               ; 1624
                                                20  11 000A8          BRB     10$                                                 ;
            53      06  AB              01          02  EF 000AA 6$:      EXTZV   #2, #1, 6(CCB), R3                                  ; 1630
                        01              00          53  CF 000B0          CASEL   R3, #0, #1                                          ;
                                    0012        0004      000B4 7$:      .WORD   8$-7$,-                                             ;
                                                                          9$-7$                                                       ;
                        09      0C  AC              03  E0 000B8 8$:      BBS     #3, LOCK_FLAGS, 9$                                  ; 1633
                                7E        00G   8F  9A 000BD          MOVZBL  #BAS$K_ILLRECLOC, -(SP)                             ; 1637
                                67              01  FB 000C1          CALLS   #1, BAS$$STOP_IO                                    ;
                                                04  11 000C4          BRB     10$                                                 ; 1633
                                52        0C    AC  D0 000C6 9$:      MOVL    LOCK_FLAGS, FLAGS                                   ; 1640
                                51              52  D0 000CA 10$:     MOVL    FLAGS, R1                                           ; 1643
                                50              56  D0 000CD          MOVL    TEMP_R11, R0                                        ;
                                    00000000G   00  16 000D0          JSB     BAS$$REC_GRFA                                       ;
                                                69  16 000D6          JSB     BAS$$CB_POP                                         ; 1647
                                                56  D5 000D8          TSTL    TEMP_R11                                            ; 1653
                                                05  13 000DA          BEQL    11$                                                 ;
                                5B              56  D0 000DC          MOVL    TEMP_R11, CCB                                       ; 1656
                                                69  16 000DF          JSB     BAS$$CB_POP                                         ; 1657
                                                    04 000E1 11$:     RET                                                         ; 1660

; Routine Size:  226 bytes,    Routine Base:  _BAS$CODE + 02BE

;  817          1661  1
```

```
;  818          1662  1
;  819          1663  1 END                        !End of module - BAS$GET
;  820          1664  1
;  821          1665  0 ELUDOM
```

```
;                                PSECT SUMMARY
;
;      Name                    Bytes                          Attributes
;
;  _BAS$CODE                     928  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         4         0          581        00:01.2
```

```
;                                        COMMAND QUALIFIERS

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

; Size:          928 code + 0 data bytes
; Run Time:          00:20.9
; Elapsed Time:      00:45.4
; Lines/CPU Min:      4770
; Lexemes/CPU-Min:  26469
; Memory Used:  159 pages
; Compilation Complete
```

BASFREE
LIS

BASEXITHA
LIS

BASFETCHD
LIS

BASFORINI
LIS

BASGETRFA
LIS

BASFETCHA
LIS

BASGET
LIS

BASFSP
LIS

BASFIND
LIS

BASFORMAT
LIS

BASHANDLE
LIS