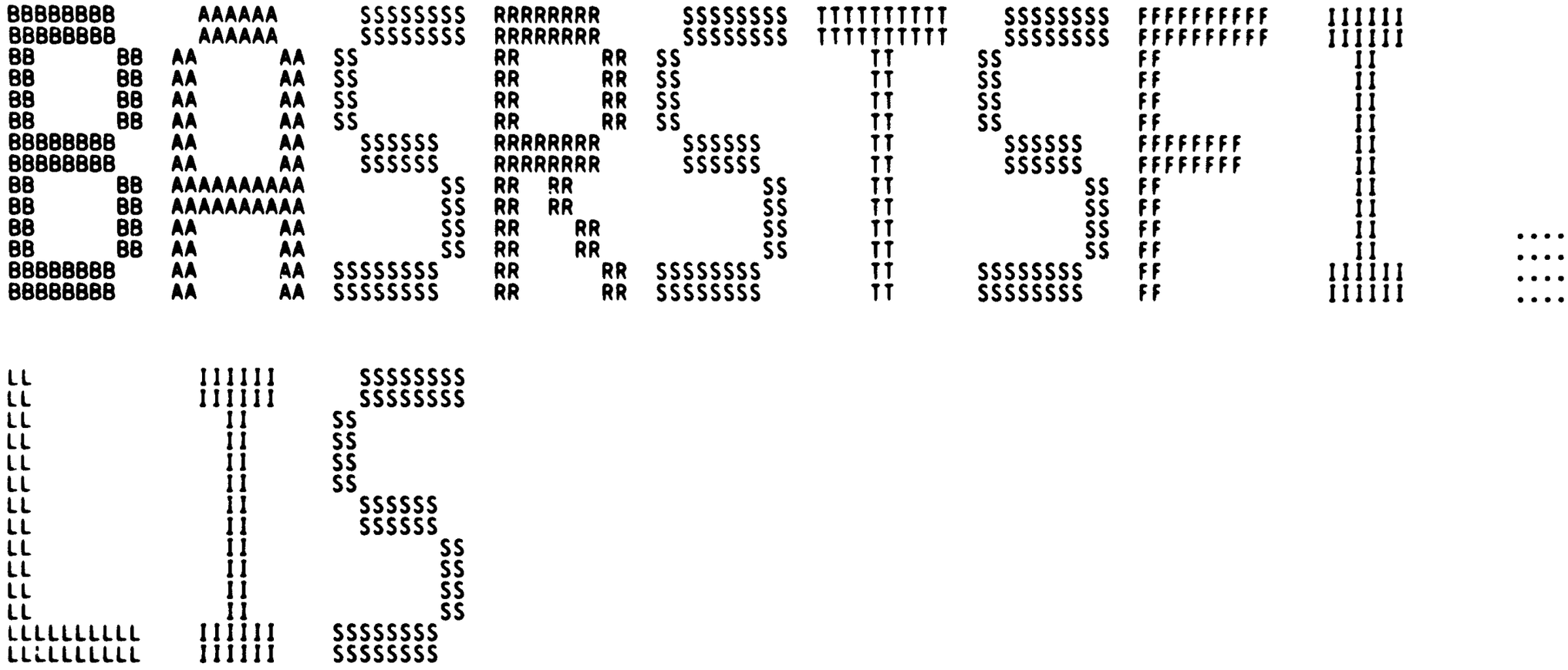
M 1

```
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
```

FILEID**BASRSTSFI



```
;     1          0001  0 MODULE BAS$RSTS_FIELD (                      ! FIELD statement
;     2          0002  0                 IDENT = '1-023'              ! File: BASRSTSFI.B32 Edit: MDL1023
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !*****************************************************************************
;     7          0007  1 !*                                                                           *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                      *
;    11          0011  1 !*                                                                           *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE     *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    17          0017  1 !*  TRANSFERRED.                                                              *
;    18          0018  1 !*                                                                           *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    21          0021  1 !*  CORPORATION.                                                              *
;    22          0022  1 !*                                                                           *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*****************************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:  VAX-11 BASIC Miscellaneous
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains the RSTS-compatable FIELD functions.
;    36          0036  1 !       A FIELD variable is semi-interpreted, and routines in this
;    37          0037  1 !       module "declare" such a variable, copy data to and from it,
;    38          0038  1 !       and purge it when the block it was in is exited.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:  VAX-11 User Mode
;    41          0041  1 !
;    42          0042  1 ! AUTHOR: John Sauter, CREATION DATE: 27-FEB-1979
;    43          0043  1 !
;    44          0044  1 ! MODIFIED BY:
;    45          0045  1 !
;    46          0046  1 ! 1-001 - Original.  JBS 27-FEB-1979
;    47          0047  1 ! 1-002 - Rearrange FIELD_SET so that the compiler can call it conveniently
;    48          0048  1 !         once for a FIELD statement.  JBS 01-MAR-1979
;    49          0049  1 ! 1-003 - Add a statement type parameter to FIELD_COPY.  JBS 02-APR-1979
;    50          0050  1 ! 1-004 - Correct STR$COPY to STR$COPY_DX.  JBS 03-APR-1979
;    51          0051  1 ! 1-005 - Re-order some parameters to make things easier on the BASIC-PLUS-2
;    52          0052  1 !         compiler.  JBS 18-MAY-1979
;    53          0053  1 ! 1-006 - Today the compiler began producing code for the FIELD
;    54          0054  1 !         statement, so begin debugging.  JBS 22-MAY-1979
;    55          0055  1 ! 1-007 - Add OPEN, CLOSE and KILL entry points.  JBS 24-MAY-1979
;    56          0056  1 ! 1-008 - Complete coding of the new entry points.  JBS 25-MAY-1979
;    57          0057  1 ! 1-009 - Add BAS$$FIELD_INIT.  JBS 04-JUN-1979
```

```
;    58          0058  1 ! 1-010 - Fix a bug in KILL which made it run forever.  JBS 07-JUN-1979
;    59          0059  1 ! 1-011 - Allow a virtual array to be fielded, but only if it is used
;    60          0060  1 !         exclusively for block I/O.  JBS 22-JUN-1979
;    61          0061  1 ! 1-012 - Add BAS$FIELD_COP_R.  JBS 13-JUL-1979
;    62          0062  1 ! 1-013 - Change calls to STR$COPY.  JBS 16-JUL-1979
;    63          0063  1 ! 1-014 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
;    64          0064  1 ! 1-015 - Call STR$FREE1_DX only for variables not already FIELDed.
;    65          0065  1 !         JBS 02-AUG-1979
;    66          0066  1 ! 1-016 - Call BAS$$CB_GET, so we don't have to be in the sharable
;    67          0067  1 !         library.  JBS 22-AUG-1979
;    68          0068  1 ! 1-017 - Make negative string lengths back up the address but set
;    69          0069  1 !         the variable's length to zero.  JBS 28-FEB-1980
;    70          0070  1 ! 1-018 - When opening a file, validate variables which may have been
;    71          0071  1 !         invalidated by the implied close.  JBS 23-MAY-1980
;    72          0072  1 ! 1-019 - Add 2 to lub$k_ilun_min before calling bas$$cb_push to force an
;    73          0073  1 !         error when -7 or -8 LUNs are used, at the same time put the code that
;    74          0074  1 !         opens channel 0 for upgrading bas$field_set to the level of the other
;    75          0075  1 !         I/O support routines.  FM 17-sep-80
;    76          0076  1 ! 1-020 - Make SYM$Q_ROOT global so that BAS$CLOSE can access it.
;    77          0077  1 !         PL 2-JUN-81
;    78          0078  1 ! 1-021 - BAS$CLOSE will not need SYM$Q_ROOT after all, so make it OWN again.
;    79          0079  1 !         PL 16-Jun-81
;    80          0080  1 ! 1-022 - Undo 19.  We can now do I/O to #0, becuase BAS$PUT will use foreign
;    81          0081  1 !         buffer mechanism to do PUTs to #0.  FM 9-JUL-81.
;    82          0082  1 ! 1-023 - set LUB$V_FIELD_USE when a FIELD statement is executed, so that
;    83          0083  1 !         BAS$CLOSE can tell if there is FIELDing on the channel.  Clear it
;    84          0084  1 !         when the channel is closed.  MDL 29-Mar-1984
;    85          0085  1 !--
;    86          0086  1
;    87          0087  1 !<BLF/PAGE>
```

```
;    89          0088  1 !
;    90          0089  1 ! SWITCHES:
;    91          0090  1 !
;    92          0091  1
;    93          0092  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    94          0093  1
;    95          0094  1 !
;    96          0095  1 ! LINKAGES:
;    97          0096  1 !
;    98          0097  1
;    99          0098  1 REQUIRE 'RTLIN:OTSLNK';                         ! Define linkages
;   100          0527  1
;   101          0528  1 !
;   102          0529  1 ! TABLE OF CONTENTS:
;   103          0530  1 !
;   104          0531  1
;   105          0532  1 FORWARD ROUTINE
;   106          0533  1     BAS$FIELD_SET : NOVALUE,                    ! Process a FIELD statement
;   107          0534  1     BAS$FIELD_VAR : CALL_CCB NOVALUE,           ! Declare a FIELD variable
;   108          0535  1     BAS$FIELD_CLEAR : NOVALUE,                  ! Undeclare a FIELD variable
;   109          0536  1     BAS$FIELD_COPY : NOVALUE,                   ! Reference such a variable
;   110          0537  1     BAS$FIELD_COP_R : NOVALUE,                  ! Reference such a variable
;   111          0538  1     BAS$FIELD_PURGE : NOVALUE,                  ! Purge field variables
;   112          0539  1     BAS$FIELD_OPEN : NOVALUE,                   ! A file was just opened
;   113          0540  1     BAS$FIELD_CLOSE : NOVALUE,                  ! A file was just closed
;   114          0541  1     BAS$$FIELD_KILL : CALL_CCB NOVALUE,         ! CLOSE appendage
;   115          0542  1     BAS$$FIELD_INIT : NOVALUE;                  ! Initialize the FIELD list
;   116          0543  1
;   117          0544  1 !
;   118          0545  1 ! INCLUDE FILES:
;   119          0546  1 !
;   120          0547  1
;   121          0548  1 REQUIRE 'RTLIN:RTLPSECT';                       ! Macros for defining psects
;   122          0643  1
;   123          0644  1 REQUIRE 'RTLML:OTSLUB';                        ! Logical unit block definitions
;   124          0784  1
;   125          0785  1 REQUIRE 'RTLML:OTSISB';                        ! ISB definitions
;   126          0953  1
;   127          0954  1 REQUIRE 'RTLIN:BASFRAME';                      ! BASIC frame structure
;   128          1157  1
;   129          1158  1 LIBRARY 'RTLSTARLE';                           ! System definitions
;   130          1159  1
;   131          1160  1 !
;   132          1161  1 ! MACROS:
;   133          1162  1 !
;   134          1163  1 !       NONE
;   135          1164  1 !
;   136          1165  1 ! EQUATED SYMBOLS:
;   137          1166  1 !
;   138          1167  1
;   139          1168  1 LITERAL
;   140          1169  1     STMT_TYPE_LSET = 0,                         ! LSET statement
;   141          1170  1     STMT_TYPE_RSET = 1;                         ! RSET statement
;   142          1171  1
;   143          1172  1 !
;   144          1173  1 ! PSECTS:
;   145          1174  1 !
```

```
;  146          1175  1 DECLARE_PSECTS (BAS);                        ! Declare psects for BAS$ facility
;  147          1176  1 !
;  148          1177  1 ! OWN STORAGE:
;  149          1178  1 !
;  150          1179  1
;  151          1180  1 OWN
;  152          1181  1     SYM$Q_ROOT : VECTOR [2] INITIAL (0, 0);     ! Root for symbol table
;  153          1182  1
;  154          1183  1 !
;  155          1184  1 ! EXTERNAL REFERENCES:
;  156          1185  1 !
;  157          1186  1
;  158          1187  1 EXTERNAL ROUTINE
;  159          1188  1     BAS$$STOP : NOVALUE,                       ! Signal fatal error
;  160          1189  1     BAS$$STOP_IO : NOVALUE,                    ! Signal fatal I/O error
;  161          1190  1     LIB$GET_VM,                                ! Get virtual memory
;  162          1191  1     LIB$FREE_VM,                               ! Free virtual memory
;  163          1192  1     BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,        ! Load register CCB
;  164          1193  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,          ! Done with register CCB
;  165          1194  1     BAS$$CB_GET : JSB_CB_GET NOVALUE,          ! Load CCB with current LUB
;  166          1195  1     STR$COPY_DX,                               ! Copy a string (LSET)
;  167          1196  1     STR$FREE1_DX,                              ! Free a string
;  168          1197  1     BAS$RSET,                                  ! Copy a string (RSET)
;  169          1198  1     BAS$$OPEN_ZERO : NOVALUE;                  ! Open channel 0
;  170          1199  1
;  171          1200  1 !+
;  172          1201  1 ! The following are the error codes used in this module.
;  173          1202  1 !-
;  174          1203  1
;  175          1204  1 EXTERNAL LITERAL
;  176          1205  1     BAS$K_MAXMEMEXC : UNSIGNED (8),            ! Maximum memory exceeded
;  177          1206  1     BAS$K_PROLOSSOR : UNSIGNED (8),            ! Program lost, sorry
;  178          1207  1     BAS$K_IO_CHANOT : UNSIGNED (8),            ! I/O channel not open
;  179          1208  1     BAS$K_ILLILLACC : UNSIGNED (8),            ! Illegal or illogical access
;  180          1209  1     BAS$K_FIEOVEBUF : UNSIGNED (8),            ! Field overflows buffer
;  181          1210  1     BAS$K_ILLFIEVAR : UNSIGNED (8),            ! Illegal FIELD variable
;  182          1211  1     BAS$K_ILLIO_CHA : UNSIGNED (8);            ! Illegal I/O channel
;  183          1212  1
;  184          1213  1 !<BLF/PAGE>
```

```
;  186          1214  1 !+
;  187          1215  1 ! The following field represents a symbol table entry.
;  188          1216  1 !-
;  189          1217  1
;  190          1218  1 FIELD
;  191          1219  1     BAS$FIELD_SYM =
;  192          1220  1         SET
;  193          1221  1         SYM$A_NEXT = [0, 0, %BPADDR, 0],          ! Next symbol table entry
;  194          1222  1         SYM$A_PREV = [%UPVAL, 0, %BPADDR, 0],     ! Previous entry
;  195          1223  1         SYM$L_CHAN = [%UPVAL*2, 0, %BPVAL, 0],    ! I/O channel
;  196          1224  1         SYM$L_OFFSET = [%UPVAL*3, 0, %BPVAL, 0],          ! Offset into I/O buffer
;  197          1225  1         SYM$L_LEN = [%UPVAL*4, 0, %BPVAL, 0],     ! Number of bytes referenced
;  198          1226  1         SYM$L_DECL = [%UPVAL*5, 0, %BPVAL, 0],    ! Scope of the declaration
;  199          1227  1         SYM$A_VAR = [%UPVAL*6, 0, %BPADDR, 0],    ! Address of descriptor
;  200          1228  1         SYM$V_INVALID = [%UPVAL*7, 0, 1, 0]       ! "Invalid" bit
;  201          1229  1         TES;
;  202          1230  1
;  203          1231  1 LITERAL
;  204          1232  1     SYM$K_LENGTH = %UPVAL*8;                      ! Number of bytes to allocate
;  205          1233  1
```

```
;  207     1234  1 GLOBAL ROUTINE BAS$FIELD_SET                  ! Execute a FIELD statement
;  208     1235  1     : NOVALUE =
;  209     1236  1
;  210     1237  1 !++
;  211     1238  1 ! FUNCTIONAL DESCRIPTION:
;  212     1239  1 !
;  213     1240  1 !       Execute a FIELD statement.  The compiler pushes all of the
;  214     1241  1 !       variables in the FIELD statement from right to left and then
;  215     1242  1 !       calls this routine.  As a result, the formal parameters are
;  216     1243  1 !       arranged rather strangely.  This routine goes through them
;  217     1244  1 !       calling BAS$FIELD_VAR for each variable.
;  218     1245  1 !
;  219     1246  1 !       The FIELD statement is formatted as follows:
;  220     1247  1 !
;  221     1248  1 !       FIELD #chan, exp BY var, exp BY var, ...
;  222     1249  1 !
;  223     1250  1 ! FORMAL PARAMETERS:
;  224     1251  1 !
;  225     1252  1 !   The formal parameters are rather strange, for the convenience of
;  226     1253  1 !   the compiler.  Because the compiler likes to push parameters in the
;  227     1254  1 !   order in which it encounters them, the pairs of optional parameters
;  228     1255  1 !   are first, followed by the fixed parameters.  The list below is of
;  229     1256  1 !   the parameters in reverse order.
;  230     1257  1 !
;  231     1258  1 !       CHAN.rl.v       An I/O channel.  Must be open.
;  232     1259  1 !       DECL.rl.v       An indication of the scope of the declaration
;  233     1260  1 !                       of the variable.  This is a pointer to the major
;  234     1261  1 !                       frame (R11) if the variable is in the scope of
;  235     1262  1 !                       the major procedure, or a pointer to the minor
;  236     1263  1 !                       frame (R10) if the variable is in the scope of
;  237     1264  1 !                       a DEF.
;  238     1265  1 !
;  239     1266  1 !   The following two parameters can be repeated as often as required.
;  240     1267  1 !
;  241     1268  1 !       LEN.rl.v        The number of bytes referenced by the variable
;  242     1269  1 !       VAR.wt.d        The variable.  Since references to it ignore its
;  243     1270  1 !                       previous (non-FIELD) contents, we free it here.
;  244     1271  1 !
;  245     1272  1 ! IMPLICIT INPUTS:
;  246     1273  1 !
;  247     1274  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  248     1275  1 !       LUB$V_VA_USE    If this bit is set, the file has been used
;  249     1276  1 !                       with a virtual array, and so cannot be used
;  250     1277  1 !                       with the FIELD statement.
;  251     1278  1 !
;  252     1279  1 ! IMPLICIT OUTPUTS:
;  253     1280  1 !
;  254     1281  1 !       SYM$Q_ROOT.mq
;  255     1282  1 !       LUB$V_BLK_USE   This bit is set to prevent the file from
;  256     1283  1 !                       being used with a virtual array.
;  257     1284  1 !       LUB$V_FIELD_USE for this channel, set to 1
;  258     1285  1 !
;  259     1286  1 ! ROUTINE VALUE:
;  260     1287  1 ! COMPLETION CODES:
;  261     1288  1 !
;  262     1289  1 !       NONE
;  263     1290  1 !
```

```
;  264          1291  1 ! SIDE EFFECTS:
;  265          1292  1 !
;  266          1293  1 !       Adds symbols to the interpreter's symbol table, and/or modifies
;  267          1294  1 !       symbols already there.
;  268          1295  1 !
;  269          1296  1 !--
;  270          1297  1
;  271          1298  2     BEGIN
;  272          1299  2
;  273          1300  2     GLOBAL REGISTER
;  274          1301  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  275          1302  2
;  276          1303  2     BUILTIN
;  277          1304  2         FP,
;  278          1305  2         ACTUALCOUNT,
;  279          1306  2         ACTUALPARAMETER;
;  280          1307  2
;  281          1308  2     LOCAL
;  282          1309  2         FMP : REF BLOCK [, BYTE],
;  283          1310  2         NUMARGS,
;  284          1311  2         DECL,
;  285          1312  2         CHAN,
;  286          1313  2         LEN,
;  287          1314  2         VAR,
;  288          1315  2         OFFSET,
;  289          1316  2         LUN_NO,
;  290          1317  2         RBF,
;  291          1318  2         RSZ;
;  292          1319  2
;  293          1320  2     FMP = .FP;
;  294          1321  2 !+
;  295          1322  2 ! Start at the beginning of the buffer.
;  296          1323  2 !-
;  297          1324  2     OFFSET = 0;
;  298          1325  2 !+
;  299          1326  2 ! Fetch, from the pecucliar argument list, the DECL and CHAN parameters.
;  300          1327  2 !-
;  301          1328  2     NUMARGS = ACTUALCOUNT ();
;  302          1329  2     CHAN = ACTUALPARAMETER (.NUMARGS);
;  303          1330  2     DECL = ACTUALPARAMETER (.NUMARGS - 1);
;  304          1331  2 !+
;  305          1332  2 ! Compute the logical unit number corresponding to th' channel, and
;  306          1333  2 ! validate it.
;  307          1334  2 !-
;  308          1335  2
;  309          1336  2     IF (.CHAN LSS 0) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
;  310          1337  2
;  311          1338  2     LUN_NO = (IF (.CHAN EQL 0) THEN LUB$K_LUN_INPU ELSE .CHAN);
;  312          1339  2     BAS$$CB_PUSH (.LUN_NO, LUB$K_ILUN_MIN);
;  313          1340  2     CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  314          1341  2
;  315          1342  3     IF ( NOT .CCB [LUB$V_OPENED])
;  316          1343  2     THEN
;  317          1344  3         IF  (.LUN_NO LSS 0)
;  318          1345  2         THEN
;  319          1346  3             BEGIN
;  320          1347  3             BAS$$OPEN_ZERO (.FMP [SF$L_SAVE_FP] );
```

```
;  321          1348  3            END
;  322          1349  2        ELSE
;  323          1350  3            BEGIN
;  324          1351  3            BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  325          1352  2            END;
;  326          1353  2
;  327          1354  2    IF (.CCB [LUB$A_CLOSE] EQLA 0) THEN CCB [LUB$A_CLOSE] = BAS$$FIELD_KILL;
;  328          1355  2
;  329          1356  2    IF (.CCB [LUB$A_CLOSE] NEQA BAS$$FIELD_KILL) THEN BAS$$STOP_IO (BAS$K_ILLILLACC);
;  330          1357  2
;  331          1358  2    CCB [LUB$V_BLK_USE] = 1;
;  332          1359  2
;  333          1360  2    CCB [LUB$V_FIELD_USE] = 1;
;  334          1361  2
;  335          1362  2    IF (.CCB [LUB$V_VA_USE]) THEN BAS$$STOP_IO (BAS$K_ILLILLACC);
;  336          1363  2
;  337          1364  2 !+
;  338          1365  2 ! Fetch the buffer address and length for later use
;  339          1366  2 !-
;  340          1367  2    RBF = .CCB [LUB$A_RBUF_ADR];
;  341          1368  2    RSZ = .CCB [LUB$W_RBUF_SIZE];
;  342          1369  2 !+
;  343          1370  2 ! Go through the optional arguments, associating each variable with
;  344          1371  2 ! its place in the I/O buffer.  We scan the variables from left to
;  345          1372  2 ! right in the FIELD statement in case the same variable appears twice:
;  346          1373  2 ! it should end up with its right-most association.
;  347          1374  2 !-
;  348          1375  2
;  349          1376  2    DECR ARGNO FROM .NUMARGS - 2 TO 1 BY 2 DO
;  350          1377  3        BEGIN
;  351          1378  3
;  352          1379  3        LOCAL
;  353          1380  3            LEN,
;  354          1381  3            VAR;
;  355          1382  3
;  356          1383  3        LEN = ACTUALPARAMETER (.ARGNO);
;  357          1384  3        VAR = ACTUALPARAMETER (.ARGNO - 1);
;  358          1385  3
;  359          1386  3        IF (.OFFSET + .LEN GTR .RSZ) THEN BAS$$STOP_IO (BAS$K_FIEOVEBUF);
;  360          1387  3
;  361          1388  3        BAS$FIELD_VAR (.CHAN, .OFFSET, .LEN, .DECL, .VAR, .RBF);
;  362          1389  3        OFFSET = .OFFSET + .LEN;
;  363          1390  2        END;
;  364          1391  2
;  365          1392  2 !+
;  366          1393  2 ! We are done with register CCB
;  367          1394  2 !-
;  368          1395  2    BAS$$CB_POP ();
;  369          1396  2    RETURN;
;  370          1397  1    END;                                    ! end of BAS$FIELD_SET

                                                                 .TITLE  BAS$RSTS_FIELD
                                                                 .IDENT  \1-023\

                                                                 .PSECT  _BAS$DATA,NOEXE,  PIC,2
```

```
                00000000  00000000  00000 SYM$Q_ROOT:
                                              .LONG   0, 0                                                    ;

                                              .EXTRN  BAS$$STOP, BAS$$STOP_IO
                                              .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                              .EXTRN  BAS$$CB_PUSH, BAS$$CB_POP
                                              .EXTRN  BAS$$CB_GET, STR$COPY_DX
                                              .EXTRN  STR$FREE1_DX, BAS$RSET
                                              .EXTRN  BAS$$OPEN_ZERO, BAS$K_MAXMEMEXC
                                              .EXTRN  BAS$K_PROCESSOR
                                              .EXTRN  BAS$K_IO_CHANOT
                                              .EXTRN  BAS$K_ILLILLACC
                                              .EXTRN  BAS$K_FIEOVEBUF
                                              .EXTRN  BAS$K_ILLFIEVAR
                                              .EXTRN  BAS$K_ILLIO_CHA

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

                                  0FFC 00000          .ENTRY  BAS$FIELD_SET, Save R2,R3,R4,R5,R6,R7,R8,-      ; 1234
                                                              R9,R10,R11
                      5A 00000000G   00  9E 00002          MOVAB   BAS$$STOP_IO, R10
                      55             5D  D0 00009          MOVL    FP, FMP                                   ; 1320
                                     54  D4 0000C          CLRL    OFFSET                                    ; 1324
                      53             6C  9A 0000E          MOVZBL  (AP), NUMARGS                             ; 1328
                      56         6C43  D0 00011          MOVL    (AP)[NUMARGS], CHAN                       ; 1329
                      59     FC AC43  D0 00015          MOVL    -4(AP)[NUMARGS], DECL                     ; 1330
                                     56  D5 0001A          TSTL    CHAN                                      ; 1336
                                     0B  18 0001C          BGEQ    1$
                      7E         00G  8F  9A 0001E          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)
            00000000G   00             01  FB 00022          CALLS   #1, BAS$$STOP
                                     56  D5 00029 1$:       TSTL    CHAN                                      ; 1338
                                     05  12 0002B          BNEQ    2$
                      52             07  CE 0002D          MNEGL   #7, LUN_NO
                                     03  11 00030          BRB     3$
                      52             56  D0 00032 2$:       MOVL    CHAN, LUN_NO
                      50             08  CE 00035 3$:       MNEGL   #8, R0                                    ; 1339
                          00000000G  00  16 00038          JSB     BAS$$CB_PUSH
                FF4C  CB         0C  A5  D0 0003E          MOVL    12(FMP), -180(CCB)                        ; 1340
                      17         FC  AB  E8 00044          BLBS    -4(CCB), 5$                               ; 1342
                                     52  D5 00048          TSTL    LUN_NO                                    ; 1344
                                     0C  18 0004A          BGEQ    4$
                                 0C  A5  DD 0004C          PUSHL   12(FMP)                                   ; 1347
            00000000G   00             01  FB 0004F          CALLS   #1, BAS$$OPEN_ZERO
                                     07  11 00056          BRB     5$                                        ; 1344
                      7E         00G  8F  9A 00058 4$:       MOVZBL  #BAS$K_IO_CHANOT, -(SP)                   ; 1351
                      6A             01  FB 0005C          CALLS   #1, BAS$$STOP_IO
                                 A4  AB  D5 0005F 5$:       TSTL    -92(CCB)                                  ; 1354
                                     06  12 00062          BNEQ    6$
                        A4   AB  0000V  CF  9E 00064          MOVAB   BAS$$FIELD_KILL, -92(CCB)
                      50     AB  0000V  CF  9E 0006A 6$:       MOVAB   BAS$$FIELD_KILL, R0                       ; 1356
                      50         A4  AB  D1 0006F          CMPL    -92(CCB), R0
                                     07  13 00073          BEQL    7$
                      7E         00G  8F  9A 00075          MOVZBL  #BAS$K_ILLILLACC, -(SP)
                      6A             01  FB 00079          CALLS   #1, BAS$$STOP_IO
                      FF     AB      02  88 0007C 7$:       BISB2   #2, -1(CCB)                               ; 1358
                      A1     AB  40  8F  88 00080          BISB2   #64, -95(CCB)                             ; 1360
                                 FF  AB  E9 00085          BLBC    -1(CCB), 8$                               ; 1362
```

```
                        7E        00    8F  9A 00089          MOVZBL  #BAS$K_ILLILLACC, -(SP)                 ;
                        6A              C1  FB 0008D          CALLS   #1, BAS$$STOP_IO                        ;
                        58        EC    AB  D0 00090 8$:      MOVL    -20(CCB), RBF                           ; 1367
                        57        D2    AB  3C 00094          MOVZWL  -46(CCB), RSZ                           ; 1368
                        52              53  D0 00098          MOVL    NUMARGS, ARGNO                          ; 1388
                                        2D  11 0009B          BRB     1$                                      ;
                        53          6C42  D0 0009D 9$:        MOVL    (AP)[ARGNO], LEN                        ; 1383
                        55        FC AC42  D0 000A1          MOVL    -4(AP)[ARGNO], VAR                      ; 1384
                50      54              53  C1 000A6          ADDL3   LEN, OFFSET, R0                         ; 1386
                        57              50  D  000AA          CMPL    R0, RSZ                                 ;
                                        07  1> 000AD          BLEQ    10$                                     ;
                        7E        00G   8F  9A 000AF          MOVZBL  #BAS$K_FIEOVEBUF, -(SP)                 ;
                        6A              01  FB 000B3          CALLS   #1, BAS$$STOP_IO                        ;
                                0120    8F  BB 000B6 10$:     PUSHR   #^M<R5,R8>                              ; 1388
                                0208    8F  BB 000BA          PUSHR   #^M<R3,R9>                              ;
                                        54  DD 000BE          PUSHL   OFFSET                                  ;
                                        56  DD 000C0          PUSHL   CHAN                                    ;
                    0000V       CF          06  FB 000C2          CALLS   #6, BAS$FIELD_VAR                   ;
                        54              53  C0 000C7          ADDL2   LEN, OFFSET                             ; 1389
                        52              02  C2 000CA 11$:     SUBL2   #2, ARGNO                               ; 1376
                                        CE  14 000CD          BGTR    9$                                      ;
                           00000000G    00  16 000CF          JSB     BAS$$CB_POP                             ; 1395
                                            04 000D5          RET                                             ; 1397

; Routine Size:  214 bytes,    Routine Base:  _BAS$CODE + 0000

;  371          1398  !
```

```
;  373          1399  1 ROUTINE BAS$FIELD_VAR (                       ! "Declare" a field variable
;  374          1400  1         CHAN,                                 ! The I/O channel whose buffer this variable references
;  375          1401  1         OFFSET,                               ! The offset in the buffer
;  376          1402  1         LEN,                                  ! The number of bytes referenced
;  377          1403  1         DECL,                                 ! Scope of the declaration
;  378          1404  1         VAR,                                  ! Address of this variable's descriptor
;  379          1405  1         RBF                                   ! Start of the record buffer
;  380          1406  1     ) : CALL_CCB NOVALUE =
;  381          1407  1
;  382          1408  1 !++
;  383          1409  1 ! FUNCTIONAL DESCRIPTION:
;  384          1410  1 !
;  385          1411  1 !       "Declares" a field variable.  Such a variable refers to the
;  386          1412  1 !       buffer of an I/O channel.  To avoid leaving obsolete addresses
;  387          1413  1 !       in a user's program each reference to a FIELD variable is
;  388          1414  1 !       interpreted.  This routine puts the variable in the interpreter
;  389          1415  1 !       symbol table so it can be found by BAS$FIELD_COPY.
;  390          1416  1 !
;  391          1417  1 ! FORMAL PARAMETERS:
;  392          1418  1 !
;  393          1419  1 !       CHAN.rl.v       An I/O channel.  Need not be open yet.
;  394          1420  1 !       OFFSET.rl.v     The offset into that channel's buffer of the
;  395          1421  1 !                       start of the area referenced by the variable
;  396          1422  1 !       LEN.rl.v        The number of bytes referenced by the variable
;  397          1423  1 !       DECL.rl.v       An indication of the scope of the declaration
;  398          1424  1 !                       of the variable.  This is a pointer to the major
;  399          1425  1 !                       frame (R11) if the variable is in the scope of
;  400          1426  1 !                       the major procedure, or a pointer to the minor
;  401          1427  1 !                       frame (R10) if the variable is in the scope of
;  402          1428  1 !                       a DEF.
;  403          1429  1 !       VAR.wt.d        The variable.  Its storage is freed and it is
;  404          1430  1 !                       made to point to the buffer, so the compiled
;  405          1431  1 !                       code can do read accesses through it.
;  406          1432  1 !       RBF.ra.v        Address of the file's record buffer.
;  407          1433  1 !
;  408          1434  1 ! IMPLICIT INPUTS:
;  409          1435  1 !
;  410          1436  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  411          1437  1 !
;  412          1438  1 ! IMPLICIT OUTPUTS:
;  413          1439  1 !
;  414          1440  1 !       SYM$Q_ROOT.mq
;  415          1441  1 !
;  416          1442  1 ! ROUTINE VALUE:
;  417          1443  1 ! COMPLETION CODES:
;  418          1444  1 !
;  419          1445  1 !       NONE
;  420          1446  1 !
;  421          1447  1 ! SIDE EFFECTS:
;  422          1448  1 !
;  423          1449  1 !       Adds a symbol to the interpreter's symbol table, or modifies one
;  424          1450  1 !       already there.
;  425          1451  1 !
;  426          1452  1 !--
;  427          1453  1
;  428          1454  2     BEGIN
;  429          1455  2
```

```
;  430          1456  2     EXTERNAL REGISTER
;  431          1457  2         CCB : REF BLOCK [, BYTE];
;  432          1458  2
;  433          1459  2     MAP
;  434          1460  2         VAR : REF BLOCK [8, BYTE];
;  435          1461  2
;  436          1462  2     LOCAL
;  437          1463  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  438          1464  2         SEARCH_DONE;
;  439          1465  2
;  440          1466  2 !+
;  441          1467  2 ! If the symbol table root has not yet been initialized, initialize it.
;  442          1468  2 !-
;  443          1469  2
;  444          1470  3     IF (.SYM$Q_ROOT [0] EQL 0)
;  445          1471  2     THEN
;  446          1472  3         BEGIN
;  447          1473  3
;  448          1474  3         LOCAL
;  449          1475  3             AST_STATUS;
;  450          1476  3
;  451          1477  3         AST_STATUS = $SETAST (ENBFLG = 0);
;  452          1478  3
;  453          1479  4         IF (.SYM$Q_ROOT [0] EQL 0)
;  454          1480  3         THEN
;  455          1481  4             BEGIN
;  456          1482  4             SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  457          1483  3             END;
;  458          1484  3
;  459          1485  3         IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  460          1486  3
;  461          1487  2         END;
;  462          1488  2
;  463          1489  2 !+
;  464          1490  2 ! Search the queue to see if this variable is already on it.
;  465          1491  2 !-
;  466          1492  2     SYM = .SYM$Q_ROOT [0];
;  467          1493  2     SEARCH_DONE = 0;
;  468          1494  2
;  469          1495  2     DO
;  470          1496  3         BEGIN
;  471          1497  3
;  472          1498  4         IF (.SYM EQLA SYM$Q_ROOT)
;  473          1499  3         THEN
;  474          1500  3             SEARCH_DONE = 1
;  475          1501  3         ELSE
;  476          1502  3
;  477          1503  4             IF (.SYM [SYM$A_VAR] EQLA .VAR)
;  478          1504  3             THEN
;  479          1505  4                 BEGIN
;  480          1506  4
;  481          1507  4                 IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP_IO (BAS$K_ILLFIEVAR);
;  482          1508  4
;  483          1509  4                 SEARCH_DONE = 3;
;  484          1510  3                 END;
;  485          1511  3
;  486          1512  3         IF ( NOT .SEARCH_DONE) THEN SYM = .SYM [SYM$A_NEXT];
```

```
;  487          1513  3
;  488          1514  3          END
;  489          1515  2      UNTIL (.SEARCH_DONE);
;  490          1516  2
;  491          1517  2      IF (.SEARCH_DONE EQL 1)
;  492          1518  2      THEN
;  493          1519  2          BEGIN
;  494          1520  3      !+
;  495          1521  3      ! We must create a symbol table entry.
;  496          1522  3      !-
;  497          1523  3
;  498          1524  3          BUILTIN
;  499          1525  3              INSQUE;
;  500          1526  3
;  501          1527  3          LOCAL
;  502          1528  3              GET_VM_STATUS,
;  503          1529  3              INSQUE_ADDR;
;  504          1530  3
;  505          1531  3          GET_VM_STATUS = LIB$GET_VM (%REF (SYM$K_LENGTH), SYM);
;  506          1532  3
;  507          1533  3          IF ( NOT .GET_VM_STATUS) THEN BAS$$STOP_IO (BAS$K_MAXMEMEXC);
;  508          1534  3
;  509          1535  3          INSQUE_ADDR = SYM$Q_ROOT [1];
;  510          1536  3          INSQUE (.SYM, ..INSQUE_ADDR);          ! Tail of queue
;  511          1537  3      !+
;  512          1538  3      ! Make sure the string is empty.
;  513          1539  3      !-
;  514          1540  3          STR$FREE1_DX (.VAR);
;  515          1541  2          END;
;  516          1542  2
;  517          1543  2      !+
;  518          1544  2      ! Fill in the symbol table entry
;  519          1545  2      !-
;  520          1546  2          SYM [SYM$L_CHAN] = .CHAN;
;  521          1547  2          SYM [SYM$L_OFFSET] = .OFFSET;
;  522          1548  2          SYM [SYM$L_LEN] = .LEN;
;  523          1549  2          SYM [SYM$L_DECL] = .DECL;
;  524          1550  2          SYM [SYM$A_VAR] = .VAR;
;  525          1551  2          SYM [SYM$V_INVALID] = 0;
;  526          1552  2          VAR [DSC$W_LENGTH] = MAX (0, .LEN);
;  527          1553  2          VAR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  528          1554  2          VAR [DSC$A_POINTER] = .RBF + .OFFSET;
;  529          1555  2          RETURN;
;  530          1556  1          END;                                    ! end of BAS$FIELD_VAR


                                                .EXTRN  SYS$SETAST

                                    007C 00000 BAS$FIELD_VAR:
                                                .WORD   Save R2,R3,R4,R5,R6                     ; 1399
                56 00000000G   00    9E 00002   MOVAB   BAS$$STOP_IO, R6                        ;
                55 00000000G   00    9E 00009   MOVAB   SYS$SETAST, R5                          ;
                54 00000000'   EF    9E 00010   MOVAB   SYM$Q_ROOT, R4                          ;
                5E             08    C2 00017   SUBL2   #8, SP                                  ;
                               64    D5 0001A   TSTL    SYM$Q_ROOT                              ; 1470
                               1D    12 0001C   BNEQ    2$                                      ;
```

```
                                7E  D4 0001E          CLRL    -(SP)                                   ; 1477
                            65  01  FB 00020          CALLS   #1, SYS$SETAST                          ;
                                64  D5 00023          TSTL    SYM$Q_ROOT                              ; 1479
                                0A  12 00025          BNEQ    1$                                      ;
                                51  64  9E 00027      MOVAB   SYM$Q_ROOT, R1                          ; 1482
                        04      A4  51  D0 0002A      MOVL    R1, SYM$Q_ROOT+4                        ;
                                64  51  D0 0002E      MOVL    R1, SYM$Q_ROOT                          ;
                                09  50  D1 00031 1$:   CMPL    AST_STATUS, #9                          ; 1485
                                    05  12 00034      BNEQ    2$                                      ;
                                    01  DD 00036      PUSHL   #1                                      ;
                                65  01  FB 00038      CALLS   #1, SYS$SETAST                          ;
                        04      AE  64  D0 0003B 2$:   MOVL    SYM$Q_ROOT, SYM                         ; 1492
                                    53  D4 0003F      CLRL    SEARCH_DONE                             ; 1493
                                52  04  AE  D0 00041 3$:   MOVL    SYM, R2                             ; 1498
                                50  64  9E 00045      MOVAB   SYM$Q_ROOT, R0                          ;
                                50  52  D1 00048      CMPL    R2, R0                                  ;
                                    05  12 0004B      BNEQ    4$                                      ;
                                53  01  D0 0004D      MOVL    #1, SEARCH_DONE                         ; 1500
                                    15  11 00050      BRB     6$                                      ;
                        14      AC  18  A2  D1 00052 4$:   CMPL    24(R2), VAR                         ; 1503
                                    0E  12 00057      BNEQ    6$                                      ;
                                07  1C  A2  E9 00059      BLBC    28(R2), 5$                          ; 1507
                                7E  00G 8F  9A 0005D      MOVZBL  #BAS$K_ILLFIEVAR, -(SP)             ;
                                66  01  FB 00061      CALLS   #1, BAS$$STOP_IO                        ;
                                53  03  D0 00064 5$:   MOVL    #3, SEARCH_DONE                         ; 1509
                                07  53  E8 00067 6$:   BLBS    SEARCH_DONE, 7$                         ; 1512
                        04      AE  62  D0 0006A      MOVL    (R2), SYM                               ;
                                D0  53  E9 0006E      BLBC    SEARCH_DONE, 3$                         ; 1515
                                01  53  D1 00071 7$:   CMPL    SEARCH_DONE, #1                         ; 1517
                                    2E  12 00074      BNEQ    9$                                      ;
                            04      AE  9F 00076      PUSHAB  SYM                                     ; 1531
                        04      AE      20 D0 00079      MOVL    #32, 4(SP)                          ;
                            04      AE  9F 0007D      PUSHAB  4(SP)                                   ;
                    00000000G   00      02 FB 00080      CALLS   #2, LIB$GET_VM                      ;
                                07      50 E8 00087      BLBS    GET_VM_STATUS, 8$                   ; 1533
                                7E  00G 8F 9A 0008A      MOVZBL  #BAS$K_MAXMEMEXC, -(SP)             ;
                                66      01 FB 0008E      CALLS   #1, BAS$$STOP_IO                    ;
                                50  04  A4 9E 00091 8$:   MOVAB   SYM$Q_ROOT+4, INSQUE_ADDR          ; 1535
                            00  B0  04  BE 0E 00095      INSQUE  @SYM, @0(INSQUE_ADDR)               ; 1536
                                    14  AC DD 0009A      PUSHL   VAR                                 ; 1540
                    00000000G   00      01 FB 0009D      CALLS   #1, STR$FREE1_DX                    ;
                                50  04  AE D0 000A4 9$:   MOVL    SYM, R0                             ; 1546
                            08  A0  04  AC 7D 000A8      MOVQ    CHAN, 8(R0)                         ;
                            10  A0  0C  AC 7D 000AD      MOVQ    LEN, 16(R0)                         ; 1548
                                51  14  AC D0 000B2      MOVL    VAR, R1                             ; 1550
                            18  A0      51 D0 000B6      MOVL    R1, 24(R0)                          ;
                            1C  A0      01 8A 000BA      BICB2   #1, 28(R0)                          ; 1551
                                50  0C  AC D0 000BE      MOVL    LEN, R0                             ; 1552
                                        02 18 000C2      BGEQ    10$                                 ;
                                        50 D4 000C4      CLRL    R0                                  ;
                                61      50 B0 000C6 10$:   MOVW    R0, (R1)                            ;
                            03  A1      01 90 000C9      MOVB    #1, 3(R1)                           ; 1553
                    04  A1      18  AC      08  AC  C1 000CD      ADDL3   OFFSET, RBF, 4(R1)          ; 1554
                                            04 000D4      RET                                         ; 1556

; Routine Size:  213 bytes,    Routine Base:  _BAS$CODE + 00D6
```

```
;  531          1557  1
```

```
;  533          1558  1 GLOBAL ROUTINE BAS$FIELD_CLEAR (                  ! Undeclare a field variable
;  534          1559  1         VAR                                       ! The variable being cleared
;  535          1560  1     ) : NOVALUE =
;  536          1561  1
;  537          1562  1 !++
;  538          1563  1 ! FUNCTIONAL DESCRIPTION:
;  539          1564  1 !
;  540          1565  1 !       Undeclare a possible FIELD variable.  This routine is called
;  541          1566  1 !       prior to any BASIC statement that causes a field variable to
;  542          1567  1 !       lose its FIELD attribute, if that variable has a FIELD
;  543          1568  1 !       statement associated with it anywhere in the program.
;  544          1569  1 !
;  545          1570  1 ! FORMAL PARAMETERS:
;  546          1571  1 !
;  547          1572  1 !       VAR.at.d        The variable.  Only the address of its
;  548          1573  1 !                       descriptor is used, to scan the symbol table.
;  549          1574  1 !
;  550          1575  1 ! IMPLICIT INPUTS:
;  551          1576  1 !
;  552          1577  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  553          1578  1 !
;  554          1579  1 ! IMPLICIT OUTPUTS:
;  555          1580  1 !
;  556          1581  1 !       SYM$Q_ROOT.mq
;  557          1582  1 !
;  558          1583  1 ! ROUTINE VALUE:
;  559          1584  1 ! COMPLETION CODES:
;  560          1585  1 !
;  561          1586  1 !       NONE
;  562          1587  1 !
;  563          1588  1 ! SIDE EFFECTS:
;  564          1589  1 !
;  565          1590  1 !       May remove a symbol from the symbol table.
;  566          1591  1 !
;  567          1592  1 !--
;  568          1593  1
;  569          1594  2     BEGIN
;  570          1595  2
;  571          1596  2     MAP
;  572          1597  2         VAR : REF BLOCK [8, BYTE];
;  573          1598  2
;  574          1599  2     LOCAL
;  575          1600  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  576          1601  2         SEARCH_DONE;
;  577          1602  2
;  578          1603  2 !+
;  579          1604  2 ! If the symbol table root has not yet been initialized, initialize it.
;  580          1605  2 !-
;  581          1606  2
;  582          1607  3     IF (.SYM$Q_ROOT [0] EQL 0)
;  583          1608  2     THEN
;  584          1609  3         BEGIN
;  585          1610  3
;  586          1611  3         LOCAL
;  587          1612  3             AST_STATUS;
;  588          1613  3
;  589          1614  3         AST_STATUS = $SETAST (ENBFLG = 0);
```

```
;  590          1615  3
;  591          1616  4          IF (.SYM$Q_ROOT [0] EQL 0)
;  592          1617  3          THEN
;  593          1618  4              BEGIN
;  594          1619  4              SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  595          1620  3              END;
;  596          1621  3
;  597          1622  3          IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  598          1623  3
;  599          1624  2          END;
;  600          1625  2
;  601          1626  2      !+
;  602          1627  2      ! Search the symbol table, removing this variable if it is present.
;  603          1628  2      !-
;  604          1629  2          SYM = .SYM$Q_ROOT [0];
;  605          1630  2          SEARCH_DONE = 0;
;  606          1631  2
;  607          1632  2          DO
;  608          1633  3              BEGIN
;  609          1634  3
;  610          1635  4              IF (.SYM EQLA SYM$Q_ROOT)
;  611          1636  3              THEN
;  612          1637  3                  SEARCH_DONE = 1
;  613          1638  3              ELSE
;  614          1639  3
;  615          1640  4                  IF (.SYM [SYM$A_VAR] EQL .VAR)
;  616          1641  3                  THEN
;  617          1642  4                      BEGIN
;  618          1643  4      !+
;  619          1644  4      ! We must delete this symbol from the symbol table.
;  620          1645  4      !-
;  621          1646  4
;  622          1647  4                      BUILTIN
;  623          1648  4                          REMQUE;
;  624          1649  4
;  625          1650  4                      LOCAL
;  626          1651  4                          FREE_VM_STATUS,
;  627          1652  4                          TEMP;
;  628          1653  4
;  629          1654  4                      IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
;  630          1655  4
;  631          1656  4                      REMQUE (.SYM, TEMP);
;  632          1657  4                      VAR [DSC$W_LENGTH] = 0;
;  633          1658  4                      VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
;  634          1659  4                      VAR [DSC$A_POINTER] = 0;
;  635          1660  4                      FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
;  636          1661  4
;  637          1662  4                      IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
;  638          1663  4
;  639          1664  4                      SEARCH_DONE = 1
;  640          1665  4                      END
;  641          1666  3                  ELSE
;  642          1667  3                      SYM = .SYM [SYM$A_NEXT];
;  643          1668  3
;  644          1669  3              END
;  645          1670  2          UNTIL (.SEARCH_DONE);
;  646          1671  2
```

```
;  647          1672  1     END;                                    ! end of BAS$FIELD_CLEAR

                                007C 00000          .ENTRY  BAS$FIELD_CLEAR, Save R2,R3,R4,R5,R6        ; 1558
                            56 00000000G   00  9E 00002          MOVAB   BAS$:STOP, R6                       ;
                            55 00000000G   00  9E 00009          MOVAB   SYS$SETAST, R5                      ;
                            54 00000000'   EF  9E 00010          MOVAB   SYM$Q_ROOT, R4                      ;
                            5E                 08  C2 00017          SUBL2   #8, SP                          ;
                                               64  D5 0001A          TSTL    SYM$Q_ROOT                      ; 1607
                                               1D  12 0001C          BNEQ    2$                              ;
                                               7E  D4 0001E          CLRL    -(SP)                           ; 1614
                            65                 01  FB 00020          CALLS   #1, SYS$SETAST                  ;
                                               64  D5 00023          TSTL    SYM$Q_ROOT                      ; 1616
                                               0A  12 00025          BNEQ    1$                              ;
                            51                 64  9E 00027          MOVAB   SYM$Q_ROOT, R1                  ; 1619
                        04  A4                 51  D0 0002A          MOVL    R1, SYM$Q_ROOT+4                ;
                            64                 51  D0 0002E          MOVL    R1, SYM$Q_ROOT                  ;
                            09                 50  D1 00031 1$:      CMPL    AST_STATUS, #9                  ; 1622
                                               05  12 00034          BNEQ    2$                              ;
                                               01  DD 00036          PUSHL   #1                              ;
                            65                 01  FB 00038          CALLS   #1, SYS$SETAST                  ;
                            52                 64  D0 0003B 2$:      MOVL    SYM$Q_ROOT, SYM                 ; 1629
                                               53  D4 0003E          CLRL    SEARCH_DONE                     ; 1630
                            50                 64  9E 00040 3$:      MOVAB   SYM$Q_ROOT, R0                  ; 1635
                            50                 52  D1 00043          CMPL    SYM, R0                         ;
                                               3E  13 00046          BEQL    5$                              ;
                        04  AC          18     A2  D1 00048          CMPL    24(SYM), VAR                    ; 1640
                                               3C  12 0004D          BNEQ    6$                              ;
                            07             1C  A2  E9 0004F          BLBC    28(SYM), 4$                     ; 1654
                            7E             00G 8F  9A 00053          MOVZBL  #BAS$K_ILLFIEVAR, -(SP)         ;
                            66                 01  FB 00057          CALLS   #1, BA$$$STOP                   ;
                        04  AE                 62  0F 0005A 4$:      REMQUE  (SYM), TEMP                     ; 1656
                            50             04  AC  D0 0005E          MOVL    VAR, R0                         ; 1657
                                               60  B4 00062          CLRW    (R0)                            ;
                        03  A0                 02  90 00064          MOVB    #2, 3(R0)                       ; 1658
                                           04  A0  D4 00068          CLRL    4(R0)                           ; 1659
                                           04  AE  9F 0006B          PUSHAB  TEMP                            ; 1660
                        04  AE                 20  D0 0006E          MOVL    #32, 4(SP)                      ;
                                           04  AE  9F 00072          PUSHAB  4(SP)                           ;
                    00000000G   00             02  FB 00075          CALLS   #2, LIB$FREE_VM                 ;
                            07                 50  E8 0007C          BLBS    FREE_VM_STATUS, 5$              ; 1662
                            7E             00G 8F  9A 0007F          MOVZBL  #BAS$K_PROLOSSOR, -(SP)         ;
                            66                 01  FB 00083          CALLS   #1, BA$$$STOP                   ;
                            53                 01  D0 00086 5$:      MOVL    #1, SEARCH_DONE                 ; 1664
                                               03  11 00089          BRB     7$                              ;
                            52                 62  D0 0008B 6$:      MOVL    (SYM), SYM                      ; 1667
                            AF                 53  E9 0008E 7$:      BLBC    SEARCH_DONE, 3$                 ; 1670
                                                   04 00091          RET                                     ; 1672

; Routine Size:  146 bytes,    Routine Base:  _BAS$CODE + 01AB

;  648          1673  1
```

```
;  650          1674  1 GLOBAL ROUTINE BAS$FIELD_COPY (                      ! Copy to or from a FIELD variable
;  651          1675  1         STMT_TYPE,                                   ! Either LSET or RSET
;  652          1676  1         VAR2,                                        ! The destination variable
;  653          1677  1         VAR1                                         ! The source variable
;  654          1678  1     ) : NOVALUE =
;  655          1679  1
;  656          1680  1 !++
;  657          1681  1 ! FUNCTIONAL DESCRIPTION:
;  658          1682  1 !
;  659          1683  1 !       Copies between two string variables.  One or the other may
;  660          1684  1 !       by FIELD variables, but not both.  Because the compiler cannot
;  661          1685  1 !       be sure if a FIELD statement has been issued to a variable
;  662          1686  1 !       (since it cannot trace program flow) it is possible that
;  663          1687  1 !       neither variable is FIELD.
;  664          1688  1 !
;  665          1689  1 ! FORMAL PARAMETERS:
;  666          1690  1 !
;  667          1691  1 !       STMT_TYPE.rl.v  0 = this is an LSET statement, 1 = RSET
;  668          1692  1 !       VAR2.wt.dx      The destination of the copy.  This may be a
;  669          1693  1 !                       FIELD variable.
;  670          1694  1 !       VAR1.rt.dx      The source for the copy.  This may be a FIELD
;  671          1695  1 !                       variable.
;  672          1696  1 ! IMPLICIT INPUTS:
;  673          1697  1 !
;  674          1698  1 !       SYM$Q_ROOT.rq   The queue of FIELD variables : the symbol table.
;  675          1699  1 !
;  676          1700  1 ! IMPLICIT OUTPUTS:
;  677          1701  1 !
;  678          1702  1 !       NONE
;  679          1703  1 !
;  680          1704  1 ! ROUTINE VALUE:
;  681          1705  1 ! COMPLETION CODES:
;  682          1706  1 !
;  683          1707  1 !       NONE
;  684          1708  1 !
;  685          1709  1 ! SIDE EFFECTS:
;  686          1710  1 !
;  687          1711  1 !       May write into or read from an I/O buffer.
;  688          1712  1 !
;  689          1713  1 !--
;  690          1714  1
;  691          1715  2     BEGIN
;  692          1716  2
;  693          1717  2     BUILTIN
;  694          1718  2         FP;
;  695          1719  2
;  696          1720  2     GLOBAL REGISTER
;  697          1721  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  698          1722  2
;  699          1723  2     MAP
;  700          1724  2         VAR1 : REF BLOCK [8, BYTE],
;  701          1725  2         VAR2 : REF BLOCK [8, BYTE];
;  702          1726  2
;  703          1727  2     LOCAL
;  704          1728  2         FMP : REF BLOCK [, BYTE],
;  705          1729  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  706          1730  2         SEARCH_DONE,
```

```
;  707          1731  2          VAR1_DESC : BLOCK [8, BYTE],
;  708          1732  2          VAR2_DESC : BLOCK [8, BYTE],
;  709          1733  2          VAR1_DESC_ADR : REF BLOCK [8, BYTE],
;  710          1734  2          VAR2_DESC_ADR : REF BLOCK [8, BYTE],
;  711          1735  2          VAR1_CHAN,
;  712          1736  2          VAR2_CHAN;
;  713          1737  2
;  714          1738  2      FMP = .FP;
;  715          1739  2  !+
;  716          1740  2  ! If the symbol table root has not yet been initialized, initialize it.
;  717          1741  2  !-
;  718          1742  2
;  719          1743  3      IF (.SYM$Q_ROOT [0] EQL 0)
;  720          1744  2      THEN
;  721          1745  3          BEGIN
;  722          1746  3
;  723          1747  3          LOCAL
;  724          1748  3              AST_STATUS;
;  725          1749  3
;  726          1750  3          AST_STATUS = $SETAST (ENBFLG = 0);
;  727          1751  3
;  728          1752  4          IF (.SYM$Q_ROOT [0] EQL 0)
;  729          1753  3          THEN
;  730          1754  4              BEGIN
;  731          1755  4              SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  732          1756  3              END;
;  733          1757  3
;  734          1758  3          IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  735          1759  3
;  736          1760  2          END;
;  737          1761  2
;  738          1762  2  !+
;  739          1763  2  ! Search the queue to see if the input variable is on it.
;  740          1764  2  !-
;  741          1765  2      SYM = .SYM$Q_ROOT [0];
;  742          1766  2      SEARCH_DONE = 0;
;  743          1767  2
;  744          1768  2      DO
;  745          1769  3          BEGIN
;  746          1770  3
;  747          1771  4          IF (.SYM EQLA SYM$Q_ROOT)
;  748          1772  3          THEN
;  749          1773  3              SEARCH_DONE = 1
;  750          1774  3          ELSE
;  751          1775  3
;  752          1776  3              IF (.SYM [SYM$A_VAR] EQLA .VAR1) THEN SEARCH_DONE = 3;
;  753          1777  3
;  754          1778  3          IF ( NOT .SEARCH_DONE) THEN SYM = .SYM [SYM$A_NEXT];
;  755          1779  3
;  756          1780  3          END
;  757          1781  2      UNTIL (.SEARCH_DONE);
;  758          1782  2
;  759          1783  3      IF (.SEARCH_DONE EQL 1)
;  760          1784  2      THEN
;  761          1785  3          BEGIN
;  762          1786  3  !+
;  763          1787  3  ! The variable is not in the symbol table.  That must mean that it
```

```
;  764          1788  3 ! is not a FIELD variable.
;  765          1789  3 !-
;  766          1790  3         VAR1_DESC_ADR = .VAR1;
;  767          1791  3         VAR1_CHAN = 0;
;  768          1792  3         END
;  769          1793  2     ELSE
;  770          1794  3         BEGIN
;  771          1795  3 !+
;  772          1796  3 ! Don't touch a variable marked invalid.
;  773          1797  3 !-
;  774          1798  3
;  775          1799  3         IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
;  776          1800  3
;  777          1801  3 !+
;  778          1802  3 ! The variable is in the symbol table.  Construct a descriptor for it.
;  779          1803  3 !-
;  780          1804  3         VAR1_DESC_ADR = VAR1_DESC;
;  781          1805  3         VAR1_DESC [DSC$W_LENGTH] = MAX (0, .SYM [SYM$L_LEN]);
;  782          1806  3         VAR1_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  783          1807  3         VAR1_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  784          1808  3         VAR1_CHAN = .SYM [SYM$L_CHAN];
;  785          1809  3
;  786          1810  3         IF (.VAR1_CHAN EQL 0) THEN VAR1_CHAN = LUB$K_LUN_INPU;
;  787          1811  3
;  788          1812  3         BAS$$CB_PUSH (.VAR1_CHAN, LUB$K_LUN_INPU);
;  789          1813  3         CCB [ISB$A_USER_FP] = .FMP [SF$[_SAVE_FP];
;  790          1814  3         VAR1_DESC [DSC$A_POINTER] = .CCB [LUB$A_RBUF_ADR] + .SYM [SYM$L_OFFSET];
;  791          1815  3
;  792          1816  3         IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP (BAS$K_IO_CHANOT);
;  793          1817  3
;  794          1818  4         IF (.CCB [LUB$W_RBUF_SIZE] LSSU .SYM [SYM$L_OFFSET] + .SYM [SYM$L_LEN])
;  795          1819  3         THEN
;  796          1820  3             BAS$$STOP_IO (BAS$K_FIEOVEBUF);
;  797          1821  3
;  798          1822  2         END;
;  799          1823  2
;  800          1824  2 !+
;  801          1825  2 ! Search the queue to see if the output variable is on it.
;  802          1826  2 !-
;  803          1827  2     SYM = .SYM$Q_ROOT [0];
;  804          1828  2     SEARCH_DONE = 0;
;  805          1829  2
;  806          1830  2     DO
;  807          1831  3         BEGIN
;  808          1832  3
;  809          1833  4         IF (.SYM EQLA SYM$Q_ROOT)
;  810          1834  3         THEN
;  811          1835  3             SEARCH_DONE = 1
;  812          1836  3         ELSE
;  813          1837  3
;  814          1838  3             IF (.SYM [SYM$A_VAR] EQLA .VAR2) THEN SEARCH_DONE = 3;
;  815          1839  3
;  816          1840  3         IF ( NOT .SEARCH_DONE) THEN SYM = .SYM [SYM$A_NEXT];
;  817          1841  3
;  818          1842  3         END
;  819          1843  2     UNTIL (.SEARCH_DONE);
;  820          1844  2
```

```
;  821          1845  3      IF (.SEARCH_DONE EQL 1)
;  822          1846  2      THEN
;  823          1847  3          BEGIN
;  824          1848  3  !+
;  825          1849  3  ! The variable is not in the symbol table.  That must mean that it
;  826          1850  3  ! is not a FIELD variable.
;  827          1851  3  !-
;  828          1852  3          VAR2_DESC_ADR = .VAR2;
;  829          1853  3          VAR2_CHAN = 0;
;  830          1854  3          END
;  831          1855  2      ELSE
;  832          1856  3          BEGIN
;  833          1857  3
;  834          1858  3          IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
;  835          1859  3
;  836          1860  3  !+
;  837          1861  3  ! The variable is in the symbol table.  Construct a descriptor for it.
;  838          1862  3  !-
;  839          1863  3          VAR2_DESC_ADR = VAR2_DESC;
;  840          1864  3          VAR2_DESC [DSC$W_LENGTH] = MAX (0, .SYM [SYM$L_LEN]);
;  841          1865  3          VAR2_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  842          1866  3          VAR2_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  843          1867  3          VAR2_CHAN = .SYM [SYM$L_CHAN];
;  844          1868  3
;  845          1869  3          IF (.VAR2_CHAN EQL 0) THEN VAR2_CHAN = LUB$K_LUN_INPU;
;  846          1870  3
;  847          1871  3          BAS$$CB_PUSH (.VAR2_CHAN, LUB$K_LUN_INPU);
;  848          1872  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  849          1873  3          VAR2_DESC [DSC$A_POINTER] = .CCB [LUB$A_RBUF_ADR] + .SYM [SYM$L_OFFSET];
;  850          1874  3
;  851          1875  3          IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP (BAS$K_IO_CHANOT);
;  852          1876  3
;  853          1877  4          IF (.CCB [LUB$W_RBUF_SIZE] LSSU .SYM [SYM$L_OFFSET] + .SYM [SYM$L_LEN])
;  854          1878  3          THEN
;  855          1879  3              BAS$$STOP_IO (BAS$K_FIEOVEBUF);
;  856          1880  3
;  857          1881  2          END;
;  858          1882  2
;  859          1883  2  !+
;  860          1884  2  ! Copy from the input variable to the output variable.
;  861          1885  2  ! We must observe the semantics of the statement type.
;  862          1886  2  !-
;  863          1887  2
;  864          1888  2      CASE .STMT_TYPE FROM STMT_TYPE_LSET TO STMT_TYPE_RSET OF
;  865          1889  2          SET
;  866          1890  2
;  867          1891  2          [STMT_TYPE_LSET] :
;  868          1892  2              STR$COPY_DX (.VAR2_DESC_ADR, .VAR1_DESC_ADR);
;  869          1893  2
;  870          1894  2          [STMT_TYPE_RSET] :
;  871          1895  2              BAS$RSET (.VAR2_DESC_ADR, .VAR1_DESC_ADR);
;  872          1896  2          TES;
;  873          1897  2
;  874          1898  2  !+
;  875          1899  2  ! Release register CCB
;  876          1900  2  !-
;  877          1901  2
```

```
;  878        1902 3      IF (.VAR2_CHAN NEQ 0)
;  879        1903 2      THEN
;  880        1904 3          BEGIN
;  881        1905 3          BAS$$CB_GET ();
;  882        1906 3          BAS$$CB_POP ();
;  883        1907 2          END;
;  884        1908 2
;  885        1909 3      IF (.VAR1_CHAN NEQ 0)
;  886        1910 2      THEN
;  887        1911 3          BEGIN
;  888        1912 3          BAS$$CB_GET ();
;  889        1913 3          BAS$$CB_POP ();
;  890        1914 2          END;
;  891        1915 2
;  892        1916 1      END;                                          ! end of BAS$FIELD_COPY


                                     OFFC 00000          .ENTRY  BAS$FIELD_COPY, Save R2,R3,R4,R5,R6,R7,R8,- ; 1674
                                                                 R9,R10,R11                                   ;
                5A 00000000G  00  9E 00002          MOVAB   SYS$SETAST, R10                               ;
                59 00000000G  00  9E 00009          MOVAB   BAS$$STOP, R9                                 ;
                58 00000000'  EF  9E 00010          MOVAB   SYM$Q_ROOT, R8                                ;
                5E            10  C2 00017          SUBL2   #16, SP                                       ;
                55            5D  D0 0001A          MOVL    FP, FMP                                       ; 1738
                              68  D5 0001D          TSTL    SYM$Q_ROOT                                    ; 1743
                              1D  12 0001F          BNEQ    2$                                            ;
                              7E  D4 00021          CLRL    -(SP)                                         ; 1750
                6A            01  FB 00023          CALLS   #1, SYS$SETAST                                ;
                              68  D5 00026          TSTL    SYM$Q_ROOT                                    ; 1752
                              0A  12 00028          BNEQ    1$                                            ;
                51            68  9E 0002A          MOVAB   SYM$Q_ROOT, R1                                ; 1755
           04   A8            51  D0 0002D          MOVL    R1, SYM$Q_ROOT+4                              ;
                68            51  D0 00031          MOVL    R1, SYM$Q_ROOT                                ;
                09            50  D1 00034 1$:      CMPL    AST_STATUS, #9                                ; 1758
                              05  12 00037          BNEQ    2$                                            ;
                              01  DD 00039          PUSHL   #1                                            ;
                6A            01  FB 0003B          CALLS   #1, SYS$SETAST                                ;
                53            68  D0 0003E 2$:      MOVL    SYM$Q_ROOT, SYM                               ; 1765
                              54  D4 00041          CLRL    SEARCH_DONE                                   ; 1766
                50            68  9E 00043 3$:      MOVAB   SYM$Q_ROOT, R0                                ; 1771
                50            53  D1 00046          CMPL    SYM, R0                                       ;
                              05  12 00049          BNEQ    4$                                            ;
                54            01  D0 0004B          MOVL    #1, SEARCH_DONE                               ; 1773
                              0A  11 0004E          BRB     5$                                            ;
           0C   AC       18   A3  D1 00050 4$:      CMPL    24(SYM), VAR1                                 ; 1776
                              03  12 00055          BNEQ    5$                                            ;
                54            03  D0 00057          MOVL    #3, SEARCH_DONE                               ;
                06            54  E8 0005A 5$:      BLBS    SEARCH_DONE, 6$                               ; 1778
                53            63  D0 0005D          MOVL    (SYM), SYM                                    ;
                E0            54  E9 00060          BLBC    SEARCH_DONE, 3$                               ; 1781
                01            54  D1 00063 6$:      CMPL    SEARCH_DONE, #1                               ; 1783
                              08  12 00066          BNEQ    7$                                            ;
                57       0C   AC  D0 00068          MOVL    VAR1, VAR1_DESC_ADR                           ; 1790
                              56  D4 0006C          CLRL    VAR1_CHAN                                     ; 1791
```

```
                                  67  11 0006E       BRB     12$                                ; 1783
                      07      1C  A3  E9 00070  7$:  BLBC    28(SYM), 8$                        ; 1799
                      7E     00G  8F  9A 00074       MOVZBL  #BAS$K_ILLFIEVAR, -(SP)
                      69          01  FB 00078       CALLS   #1, BAS$$STOP
                      57      08  AE  9E 0007B  8$:  MOVAB   VAR1_DESC, VAR1_DESC_ADR           ; 1804
                      50      10  A3  D0 0007F       MOVL    16(SYM), R0                        ; 1805
                                  02  18 00083       BGEQ    9$
                                  50  D4 00085       CLRL    R0
                 08   AE          50  B0 00087  9$:  MOVW    R0, VAR1_DESC
                 0A   AE    010E  8F  B0 0008B       MOVW    #270, VAR1_DESC+2                  ; 1806
                      56      08  A3  D0 00091       MOVL    8(SYM), VAR1_CHAN                  ; 1808
                                  03  12 00095       BNEQ    10$                                ; 1810
                      56          07  CE 00097       MNEGL   #7, VAR1_CHAN
                      50          07  CE 0009A 10$:  MNEGL   #7, R0                             ; 1812
                      52          56  D0 0009D       MOVL    VAR1_CHAN, R2
                       000000000G 00  16 000A0       JSB     BAS$$CB_PUSH
               FF4C   CB      0C  A5  D0 000A6       MOVL    12(FMP), -180(CCB)                 ; 1813
      0C   AE    EC   AB      0C  A3  C1 000AC       ADDL3   12(SYM), -20(CCB), VAR1_DESC+4     ; 1814
                      07      FC  AB  E8 000B3       BLBS    -4(CCB), 11$                       ; 1816
                      7E     00G  8F  9A 000B7       MOVZBL  #BAS$K_IO_CHANOT, -(SP)
                      69          01  FB 000BB       CALLS   #1, BAS$$STOP
           50    0C   A3      10  A3  C1 000BE 11$:  ADDL3   16(SYM), 12(SYM), R0               ; 1818
 50    D2  AB         10          00  ED 000C4       CMPZV   #0, #16, -46(CCB), R0
                                  0B  1E 000CA       BGEQU   12$
                      7E     00G  8F  9A 000CC       MOVZBL  #BAS$K_FIEOVEBUF, -(SP)            ; 1820
          00000000G   00          01  FB 000D0       CALLS   #1, BAS$$STOP_IO
                      53          68  D0 000D7 12$:  MOVL    SYM$Q_ROOT, SYM                    ; 1827
                                  54  D4 000DA       CLRL    SEARCH_DONE                        ; 1828
                      50          68  9E 000DC 13$:  MOVAB   SYM$Q_ROOT, R0                     ; 1833
                      50          53  D1 000DF       CMPL    SYM, R0
                                  05  12 000E2       BNEQ    14$
                      54          01  D0 000E4       MOVL    #1, SEARCH_DONE                    ; 1835
                                  0A  11 000E7       BRB     15$
                 08   AC      18  A3  D1 000E9 14$:  CMPL    24(SYM), VAR2                      ; 1838
                                  03  12 000EE       BNEQ    15$
                      54          03  D0 000F0       MOVL    #3, SEARCH_DONE
                      06          54  E8 000F3 15$:  BLBS    SEARCH_DONE, 16$                   ; 1840
                      53          63  D0 000F6       MOVL    (SYM), SYM
                      E0          54  E9 000F9       BLBC    SEARCH_DONE, 13$                   ; 1843
                      01          54  D1 000FC 16$:  CMPL    SEARCH_DONE, #1                    ; 1845
                                  08  12 000FF       BNEQ    17$
                      54      08  AC  D0 00101       MOVL    VAR2, VAR2_DESC_ADR                ; 1852
                                  52  D4 00105       CLRL    VAR2_CHAN                          ; 1853
                                  62  11 00107       BRB     22$                                ; 1845
                      07      1C  A3  E9 00109 17$:  BLBC    28(SYM), 18$                       ; 1858
                      7E     00G  8F  9A 0010D       MOVZBL  #BAS$K_ILLFIEVAR, -(SP)
                      69          01  FB 00111       CALLS   #1, BAS$$STOP
                      54          6E  9E 00114 18$:  MOVAB   VAR2_DESC, VAR2_DESC_ADR           ; 1863
                      50      10  A3  D0 00117       MOVL    16(SYM), R0                        ; 1864
                                  02  18 0011B       BGEQ    19$
                                  50  D4 0011D       CLRL    R0
                      6E          50  B0 0011F 19$:  MOVW    R0, VAR2_DESC
                 02   AE    010E  8F  B0 00122       MOVW    #270, VAR2_DESC+2                  ; 1865
                      52      08  A3  D0 00128       MOVL    8(SYM), VAR2_CHAN                  ; 1867
                                  03  12 0012C       BNEQ    20$                                ; 1869
                      52          07  CE 0012E       MNEGL   #7, VAR2_CHAN
                      50          07  CE 00131 20$:  MNEGL   #7, R0                             ; 1871
```

```
                                    000000000G  00  16 00134          JSB     BAS$$CB_PUSH                                    ;
                          FF4C      CB          0C  A5 D0 0013A        MOVL    12(FMP), -180(CCB)                              ; 1872
                    04    AE        EC          AB  0C A3 C1 00140     ADDL3   12(SYM), -20(CCB), VAR2_DESC+4                  ; 1873
                                                0'  FC AB E8 00147     BLBS    -4(CCB), 21$                                    ; 1875
                                                7E  00G 8F 9A 0014B    MOVZBL  #BAS$K_IO_CHANOT, -(SP)                         ;
                                                69  01 FB 0014F        CALLS   #1, BAS$$STOP                                   ;
                          53        0C          A3  10 A3 C1 00152 21$: ADDL3   16(SYM), 12(SYM), R3                            ; 1877
                53        D2        AB          10  00 ED 00158        CMPZV   #0, #16, -46(CCB), R3                           ;
                                                0B  1E 0015E           BGEQU   22$                                             ;
                                                7E  00G 8F 9A 00160    MOVZBL  #BAS$K_FIEOVEBUF, -(SP)                         ; 1879
                                    000000000G  00  01 FB 00164        CALLS   #1, BAS$$STOP_IO                                ;
                                    01          00  04 AC CF 0016B 22$: CASEL   STMT_TYPE, #0, #1                               ; 1888
                                              0011      0004     00170 23$: .WORD   24$-23$,-                                    ;
                                                                            25$-23$                                           ;
                                                0090    8F BB 00174 24$:    PUSHR   #^M<R4,R7>                                  ; 1892
                                    000000000G  00      02 FB 00178        CALLS   #2, STR$COPY_DX                             ;
                                                        0B 11 0017F        BRB     26$                                         ;
                                                0090    8F BB 00181 25$:    PUSHR   #^M<R4,R7>                                  ; 1895
                                    000000000G  00      02 FB 00185        CALLS   #2, BAS$RSET                                ;
                                                        52 D5 0018C 26$:    TSTL    VAR2_CHAN                                   ; 1902
                                                        0C 13 0018E        BEQL    27$                                         ;
                                    000000000G  00      16 00190           JSB     BAS$$CB_GET                                 ; 1905
                                    000000000G  00      16 00196           JSB     BAS$$CB_POP                                 ; 1906
                                                        56 D5 0019C 27$:    TSTL    VAR1_CHAN                                   ; 1909
                                                        0C 13 0019E        BEQL    28$                                         ;
                                    000000000G  00      16 001A0           JSB     BAS$$CB_GET                                 ; 1912
                                    000000000G  00      16 001A6           JSB     BAS$$CB_POP                                 ; 1913
                                                        04 001AC 28$:      RET                                                 ; 1916

; Routine Size:  429 bytes,    Routine Base:  _BAS$CODE + 023D

;  893          1917  1
```

```
;  895          1918  1 GLOBAL ROUTINE BAS$FIELD_COP_R (                      ! Copy to a FIELD variable
;  896          1919  1         STMT_TYPE,                                    ! Either LSET or RSET
;  897          1920  1         VAR2,                                         ! The destination variable
;  898          1921  1         VAR1_LEN,                                     ! Length of the source variable
;  899          1922  1         VAR1_ADDR                                     ! Address of the source variable
;  900          1923  1     ) : NOVALUE =
;  901          1924  1
;  902          1925  1 !++
;  903          1926  1 ! FUNCTIONAL DESCRIPTION:
;  904          1927  1 !
;  905          1928  1 !       This is an alternate entry point for BAS$FIELD_COPY, which the
;  906          1929  1 !       compiler uses to avoid having to build a descriptor for a string
;  907          1930  1 !       constant.  This code builds the descriptor and calls BAS$FIELD_COPY.
;  908          1931  1 !
;  909          1932  1 ! FORMAL PARAMETERS:
;  910          1933  1 !
;  911          1934  1 !       STMT_TYPE.rl.v  0 = this is an LSET statement, 1 = RSET
;  912          1935  1 !       VAR2.wt.dx      The destination of the copy.  This may be a
;  913          1936  1 !                       FIELD variable.
;  914          1937  1 !       VAR1_LEN.rl.v   The number of bytes in the source
;  915          1938  1 !       VAR1_ADDR.rt.r  The address of the source
;  916          1939  1 !
;  917          1940  1 ! IMPLICIT INPUTS:
;  918          1941  1 !
;  919          1942  1 !       SYMSQ_ROOT.rq   The queue of FIELD variables : the symbol table.
;  920          1943  1 !
;  921          1944  1 ! IMPLICIT OUTPUTS:
;  922          1945  1 !
;  923          1946  1 !       NONE
;  924          1947  1 !
;  925          1948  1 ! ROUTINE VALUE:
;  926          1949  1 ! COMPLETION CODES:
;  927          1950  1 !
;  928          1951  1 !       NONE
;  929          1952  1 !
;  930          1953  1 ! SIDE EFFECTS:
;  931          1954  1 !
;  932          1955  1 !       May write into or read from an I/O buffer.
;  933          1956  1 !
;  934          1957  1 !--
;  935          1958  1
;  936          1959  2     BEGIN
;  937          1960  2
;  938          1961  2     LOCAL
;  939          1962  2         VAR1 : BLOCK [8, BYTE];                     ! Build source descriptor here
;  940          1963  2
;  941          1964  2     VAR1 [DSC$W_LENGTH] = .VAR1_LEN;
;  942          1965  2     VAR1 [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  943          1966  2     VAR1 [DSC$B_CLASS] = DSC$K_CLASS_S;
;  944          1967  2     VAR1 [DSC$A_POINTER] = .VAR1_ADDR;
;  945          1968  2 !+
;  946          1969  2 ! Now do the copy.
;  947          1970  2 !-
;  948          1971  2     BAS$FIELD_COPY (.STMT_TYPE, .VAR2, VAR1);
;  949          1972  1     END;                                            ! end of BAS$FIELD_COP_R
```

```
                                 0000 00000        .ENTRY  BAS$FIELD_COP_R, Save nothing                ; 1918
                          08     C2 00002        SUBL2   #8, SP                                       ;
                      0C  AC     B0 00005        MOVW    VAR1_LEN, VAR1                               ; 1964
                010E  8F     B0 00009        MOVW    #270, VAR1+2                                 ; 1965
                  10  AC     D0 0000F        MOVL    VAR1_ADDR, VAR1+4                            ; 1967
                      5E     DD 00014        PUSHL   SP                                           ; 1971
                04  AC     7D 00016        MOVQ    STMT_TYPE, -(SP)                             ;
            FE34  CF     03  FB 0001A        CALLS   #3, BAS$FIELD_COPY                           ;
                              04 0001F        RET                                                  ; 1972

; Routine Size:  32 bytes,    Routine Base:  _BAS$CODE + 03EA

;  950          1973  1
```

```
;  952          1974  1 GLOBAL ROUTINE BAS$FIELD_PURGE (                      ! Purge field variables
;  953          1975  1         DECL                                          ! Scope of the declaration
;  954          1976  1     ) : NOVALUE =
;  955          1977  1
;  956          1978  1 !++
;  957          1979  1 ! FUNCTIONAL DESCRIPTION:
;  958          1980  1 !
;  959          1981  1 !       Purge, or undeclare, field variables.  This routine is called
;  960          1982  1 !       at the end of a block with declarations that might have been
;  961          1983  1 !       FIELD variables.  It scans the symbol table and purges each
;  962          1984  1 !       entry marked as declared in the block.
;  963          1985  1 !
;  964          1986  1 ! FORMAL PARAMETERS:
;  965          1987  1 !
;  966          1988  1 !       DECL.rl.v       An indication of the scope of the declaration
;  967          1989  1 !                       of the variable.  This is a pointer to the major
;  968          1990  1 !                       frame (R11) if the variable is in the scope of
;  969          1991  1 !                       the major procedure, or a pointer to the minor
;  970          1992  1 !                       frame (R10) if the variable is in the scope of
;  971          1993  1 !                       a DEF.
;  972          1994  1 !
;  973          1995  1 ! IMPLICIT INPUTS:
;  974          1996  1 !
;  975          1997  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  976          1998  1 !
;  977          1999  1 ! IMPLICIT OUTPUTS:
;  978          2000  1 !
;  979          2001  1 !       SYM$Q_ROOT.mq
;  980          2002  1 !
;  981          2003  1 ! ROUTINE VALUE:
;  982          2004  1 ! COMPLETION CODES:
;  983          2005  1 !
;  984          2006  1 !       NONE
;  985          2007  1 !
;  986          2008  1 ! SIDE EFFECTS:
;  987          2009  1 !
;  988          2010  1 !       May remove symbols from the symbol table.
;  989          2011  1 !
;  990          2012  1 !--
;  991          2013  1
;  992          2014  2     BEGIN
;  993          2015  2
;  994          2016  2     LOCAL
;  995          2017  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  996          2018  2         SEARCH_DONE;
;  997          2019  2
;  998          2020  2 !+
;  999          2021  2 ! If the symbol table root has not yet been initialized, iritialize it.
; 1000          2022  2 !-
; 1001          2023  2
; 1002          2024  3     IF (.SYM$Q_ROOT [0] EQL 0)
; 1003          2025  2     THEN
; 1004          2026  3         BEGIN
; 1005          2027  3
; 1006          2028  3         LOCAL
; 1007          2029  3             AST_STATUS;
; 1008          2030  3
```

```
;  1009          2031  3          AST_STATUS = $SETAST (ENBFLG = 0);
;  1010          2032  3
;  1011          2033  4          IF (.SYM$Q_ROOT [0] EQL 0)
;  1012          2034  3          THEN
;  1013          2035  4              BEGIN
;  1014          2036  4              SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  1015          2037  4              END;
;  1016          2038  3
;  1017          2039  3          IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  1018          2040  3
;  1019          2041  2          END;
;  1020          2042  2
;  1021          2043  2      !+
;  1022          2044  2      ! Search the queue, removing any variables declared in this block.
;  1023          2045  2      !-
;  1024          2046  2          SYM = .SYM$Q_ROOT [0];
;  1025          2047  2          SEARCH_DONE = 0;
;  1026          2048  2
;  1027          2049  2          DO
;  1028          2050  3              BEGIN
;  1029          2051  3
;  1030          2052  4              IF (.SYM EQLA SYM$Q_ROOT)
;  1031          2053  3              THEN
;  1032          2054  3                  SEARCH_DONE = 1
;  1033          2055  3              ELSE
;  1034          2056  3
;  1035          2057  4                  IF (.SYM [SYM$L_DECL] EQL .DECL)
;  1036          2058  3                  THEN
;  1037          2059  4                      BEGIN
;  1038          2060  4      !+
;  1039          2061  4      ! We must delete this symbol from the symbol table.
;  1040          2062  4      !-
;  1041          2063  4
;  1042          2064  4                      BUILTIN
;  1043          2065  4                          REMQUE;
;  1044          2066  4
;  1045          2067  4                      LOCAL
;  1046          2068  4                          FREE_VM_STATUS,
;  1047          2069  4                          TEMP,
;  1048          2070  4                          VAR : REF BLOCK [8, BYTE];
;  1049          2071  4
;  1050          2072  4                      REMQUE (.SYM, TEMP);
;  1051          2073  4
;  1052          2074  4                      IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
;  1053          2075  4
;  1054          2076  4                      VAR = .SYM [SYM$A_VAR];
;  1055          2077  4                      VAR [DSC$W_LENGTH] = 0;
;  1056          2078  4                      VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
;  1057          2079  4                      VAR [DSC$A_POINTER] = 0;
;  1058          2080  4                      FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
;  1059          2081  4
;  1060          2082  4                      IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
;  1061          2083  4
;  1062          2084  4                      SYM = .SYM$Q_ROOT [0];
;  1063          2085  4                      END
;  1064          2086  3                  ELSE
;  1065          2087  3                      SYM = .SYM [SYM$A_NEXT];
```

```
;  1066          2088  3
;  1067          2089  3          END
;  1068          2090  2      UNTIL (.SEARCH_DONE);
;  1069          2091  2
;  1070          2092  1      END;                                    ! end of BAS$FIELD_PURGE


                                 007C 00000          .ENTRY  BAS$FIELD_PURGE, Save R2,R3,R4,R5,R6        ; 1974
                      56 00000000G   00   9E 00002   MOVAB   BAS$$STOP, R6                               ;
                      55 00000000G   00   9E 00009   MOVAB   SYS$SETAST, R5                             ;
                      54 00000000'   EF   9E 00010   MOVAB   SYM$Q_ROOT, R4                            ;
                      5E             08   C2 00017   SUBL2   #8, SP                                    ;
                                     64   D5 0001A   TSTL    SYM$Q_ROOT                                  ; 2024
                                     1D   12 0001C   BNEQ    2$                                          ;
                                     7E   D4 0001E   CLRL    -(SP)                                       ; 2031
                      65             01   FB 00020   CALLS   #1, SYS$SETAST                            ;
                                     64   D5 00023   TSTL    SYM$Q_ROOT                                  ; 2033
                                     0A   12 00025   BNEQ    1$                                          ;
                      51             64   9E 00027   MOVAB   SYM$Q_ROOT, R1                            ; 2036
                04    A4             51   D0 0002A   MOVL    R1, SYM$Q_ROOT+4                            ;
                      64             51   D0 0002E   MOVL    R1, SYM$Q_ROOT                              ;
                      09             50   D1 00031 1$:  CMPL    AST_STATUS, #9                          ; 2039
                                     05   12 00034   BNEQ    2$                                          ;
                                     01   DD 00036   PUSHL   #1                                         ;
                      65             01   FB 00038   CALLS   #1, SYS$SETAST                            ;
                      52             64   D0 0003B 2$:  MOVL    SYM$Q_ROOT, SYM                         ; 2046
                                     53   D4 0003E   CLRL    SEARCH_DONE                                 ; 2047
                      50             64   9E 00040 3$:  MOVAB   SYM$Q_ROOT, R0                         ; 2052
                      50             52   D1 00043   CMPL    SYM, R0                                     ;
                                     05   12 00046   BNEQ    4$                                          ;
                      53             01   D0 00048   MOVL    #1, SEARCH_DONE                             ; 2054
                                     46   11 0004B   BRB     8$                                          ;
                04    AC       14    A2   D1 0004D 4$:  CMPL    20(SYM), DECL                           ; 2057
                                     3C   12 00052   BNEQ    7$                                          ;
                04    AE             62   0F 00054   REMQUE  (SYM), TEMP                               ; 2072
                      07       1C    A2   E9 00058   BLBC    28(SYM), 5$                                 ; 2074
                      7E       00G   8F   9A 0005C   MOVZBL  #BAS$K_ILLFIEVAR, -(SP)                     ;
                      66             01   FB 00060   CALLS   #1, BAS$$STOP                             ;
                      50       18    A2   D0 00063 5$:  MOVL    24(SYM), VAR                            ; 2076
                                     60   B4 00067   CLRW    (VAR)                                       ; 2077
                03    A0             02   90 00069   MOVB    #2, 3(VAR)                                  ; 2078
                                04   A0   D4 0006D   CLRL    4(VAR)                                      ; 2079
                                04   AE   9F 00070   PUSHAB  TEMP                                       ; 2080
                04    AE             20   D0 00073   MOVL    #32, 4(SP)                                  ;
                                04   AE   9F 00077   PUSHAB  4(SP)                                      ;
00000000G       00                   02   FB 0007A   CALLS   #2, LIB$FREE_VM                            ;
                      07             50   E8 00081   BLBS    FREE_VM_STATUS, 6$                          ; 2082
                      7E       00G   8F   9A 00084   MOVZBL  #BAS$K_PROLOSSOR, -(SP)                     ;
                      66             01   FB 00088   CALLS   #1, BAS$$STOP                             ;
                      52             64   D0 0008B 6$:  MOVL    SYM$Q_ROOT, SYM                         ; 2084
                                     03   11 0008E   BRB     8$                                          ; 2057
                      52             62   D0 00090 7$:  MOVL    (SYM), SYM                              ; 2087
                      AA             53   E9 00093 8$:  BLBC    SEARCH_DONE, 3$                         ; 2090
                                          04 00096   RET                                                 ; 2092
```

; Routine Size: 151 bytes,   Routine Base: _BAS$CODE + 040A

; 1071   2093 1

```
;  1073          2094  1 GLOBAL ROUTINE BAS$FIELD_OPEN (                       ! Account for OPENing a file
;  1074          2095  1         CHAN                                          ! Channel just OPENed
;  1075          2096  1     ) : NOVALUE =
;  1076          2097  1
;  1077          2098  1 !++
;  1078          2099  1 ! FUNCTIONAL DESCRIPTION:
;  1079          2100  1 !
;  1080          2101  1 !       Account for OPENing a file.  If the record length is shorter
;  1081          2102  1 !       than before, some variables may have to be un-fielded.
;  1082          2103  1 !
;  1083          2104  1 ! FORMAL PARAMETERS:
;  1084          2105  1 !
;  1085          2106  1 !       CHAN.rl.v       The channel number of the file just opened.
;  1086          2107  1 !
;  1087          2108  1 ! IMPLICIT INPUTS:
;  1088          2109  1 !
;  1089          2110  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  1090          2111  1 !
;  1091          2112  1 ! IMPLICIT OUTPUTS:
;  1092          2113  1 !
;  1093          2114  1 !       SYM$Q_ROOT.mq
;  1094          2115  1 !
;  1095          2116  1 ! ROUTINE VALUE:
;  1096          2117  1 ! COMPLETION CODES:
;  1097          2118  1 !
;  1098          2119  1 !       NONE
;  1099          2120  1 !
;  1100          2121  1 ! SIDE EFFECTS:
;  1101          2122  1 !
;  1102          2123  1 !       May remove symbols from the symbol table.
;  1103          2124  1 !
;  1104          2125  1 !--
;  1105          2126  1
;  1106          2127  2     BEGIN
;  1107          2128  2
;  1108          2129  2     BUILTIN
;  1109          2130  2         FP;
;  1110          2131  2
;  1111          2132  2     GLOBAL REGISTER
;  1112          2133  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  1113          2134  2
;  1114          2135  2     LOCAL
;  1115          2136  2         FMP : REF BLOCK [, BYTE],
;  1116          2137  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  1117          2138  2         SEARCH_DONE,
;  1118          2139  2         LUN_NO,
;  1119          2140  2         RSZ,
;  1120          2141  2         RBF;
;  1121          2142  2
;  1122          2143  2     FMP = .FP;
;  1123          2144  2 !+
;  1124          2145  2 ! Compute the logical unit number from the channel number.
;  1125          2146  2 !-
;  1126          2147  2
;  1127          2148  2     IF (.CHAN LSS 0) THEN BAS$$STOP (BAS$K_ILLIO_CHA);
;  1128          2149  2
;  1129          2150  2     IF (.CHAN EQL 0) THEN LUN_NO = LUB$K_LUN_INPU ELSE LUN_NO = .CHAN;
```

```
;  1130          2151  2
;  1131          2152  2 !+
;  1132          2153  2 ! If the symbol table root has not yet been initialized, initialize it.
;  1133          2154  2 !-
;  1134          2155  2
;  1135          2156  3     IF (.SYM$Q_ROOT [0] EQL 0)
;  1136          2157  2     THEN
;  1137          2158  3         BEGIN
;  1138          2159  3
;  1139          2160  3         LOCAL
;  1140          2161  3             AST_STATUS;
;  1141          2162  3
;  1142          2163  3         AST_STATUS = $SETAST (ENBFLG = 0);
;  1143          2164  3
;  1144          2165  4         IF (.SYM$Q_ROOT [0] EQL 0)
;  1145          2166  3         THEN
;  1146          2167  4             BEGIN
;  1147          2168  4             SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  1148          2169  3             END;
;  1149          2170  3
;  1150          2171  3         IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  1151          2172  3
;  1152          2173  2         END;
;  1153          2174  2
;  1154          2175  2 !+
;  1155          2176  2 ! Pick up the buffer size to compare against the variables.
;  1156          2177  2 !-
;  1157          2178  2     BAS$$CB_PUSH (.LUN_NO, LUB$K_LUN_INPU);
;  1158          2179  2     CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  1159          2180  2     RBF = .CCB [LUB$A_RBUF_ADR];
;  1160          2181  2     RSZ = .CCB [LUB$W_RBUF_SIZE];
;  1161          2182  2 !+
;  1162          2183  2 ! Search the queue, removing any variables which no longer fit in
;  1163          2184  2 ! the current buffer.
;  1164          2185  2 !-
;  1165          2186  2     SYM = .SYM$Q_ROOT [0];
;  1166          2187  2     SEARCH_DONE = 0;
;  1167          2188  2
;  1168          2189  2     DO
;  1169          2190  3         BEGIN
;  1170          2191  3
;  1171          2192  4         IF (.SYM EQLA SYM$Q_ROOT)
;  1172          2193  3         THEN
;  1173          2194  3             SEARCH_DONE = 1
;  1174          2195  3         ELSE
;  1175          2196  3
;  1176          2197  4             IF (.SYM [SYM$L_CHAN] EQL .CHAN)
;  1177          2198  3             THEN
;  1178          2199  4                 BEGIN
;  1179          2200  4
;  1180          2201  5                 IF (.SYM [SYM$L_OFFSET] + .SYM [SYM$L_LEN] LEQ .RSZ)
;  1181          2202  4                 THEN
;  1182          2203  5                     BEGIN
;  1183          2204  5 !+
;  1184          2205  5 ! The variable is still within the buffer, recompute its address,
;  1185          2206  5 ! since the buffer may have been reallocated.
;  1186          2207  5 !-
```

```
;  1187          2208  5
;  1188          2209  5                        LOCAL
;  1189          2210  5                            VAR : REF BLOCK [8, BYTE];
;  1190          2211  5
;  1191          2212  5                        VAR = .SYM [SYM$A_VAR];
;  1192          2213  5                        VAR [DSC$A_POINTER] = .RBF + .SYM [SYM$L_OFFSET];
;  1193          2214  5 !+
;  1194          2215  5 ! Clear the "invalid" bit, since it may have been set by an implied close.
;  1195          2216  5 !-
;  1196          2217  5                        SYM [SYM$V_INVALID] = 0;
;  1197          2218  5                        SYM = .SYM [SYM$A_NEXT];
;  1198          2219  5                        END
;  1199          2220  4                    ELSE
;  1200          2221  5                        BEGIN
;  1201          2222  5 !+
;  1202          2223  5 ! This variable is outside the new buffer, remove it.
;  1203          2224  5 !-
;  1204          2225  5
;  1205          2226  5                        BUILTIN
;  1206          2227  5                            REMQUE;
;  1207          2228  5
;  1208          2229  5                        LOCAL
;  1209          2230  5                            FREE_VM_STATUS,
;  1210          2231  5                            TEMP,
;  1211          2232  5                            VAR : REF BLOCK [8, BYTE];
;  1212          2233  5
;  1213          2234  5                        REMQUE (.SYM, TEMP);
;  1214          2235  5                        VAR = .SYM [SYM$A_VAR];
;  1215          2236  5                        VAR [DSC$W_LENGTH] = 0;
;  1216          2237  5                        VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
;  1217          2238  5                        VAR [DSC$A_POINTER] = 0;
;  1218          2239  5                        FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
;  1219          2240  5
;  1220          2241  5                        IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
;  1221          2242  5
;  1222          2243  5                        SYM = .SYM$Q_ROOT [0];
;  1223          2244  5                        END
;  1224          2245  5
;  1225          2246  4                    END
;  1226          2247  3                ELSE
;  1227          2248  3                    SYM = .SYM [SYM$A_NEXT];
;  1228          2249  3
;  1229          2250  3            END
;  1230          2251  2        UNTIL (.SEARCH_DONE);
;  1231          2252  2
;  1232          2253  2 !+
;  1233          2254  2 ! We are through with register CCB.
;  1234          2255  2 !-
;  1235          2256  2        BAS$$CB_POP ();
;  1236          2257  1        END;                                    ! end of BAS$FIELD_OPEN


                                                09FC 00000          .ENTRY  BAS$FIELD_OPEN, Save R2,R3,R4,R5,R6,R7,R8,- ; 2094
                                                                            R11                                          ;
```

```
                58 00000000G   00  9E 00002          MOVAB   SYS$SETAST, R8                          ;
                57 00000000G   00  9E 00009          MOVAB   BAS$$STOP, R7                          ;
                56 00000000'   EF  9E 00010          MOVAB   SYM$Q_ROOT, R6                         ;
                5E             08  C2 00017          SUBL2   #8, SP                                 ;
                53             5D  D0 0001A          MOVL    FP, FMP                                ; 2143
                52       04    AC  D0 0001D          MOVL    CHAN, R2                               ; 2148
                                07  18 00021          BGEQ    1$                                     ;
                7E       00G   8F  9A 00023          MOVZBL  #BAS$K_ILLIO_CHA, -(SP)                ;
                67             01  FB 00027          CALLS   #1, BAS$$STOP                          ;
                               52  D5 0002A 1$:      TSTL    R2                                     ; 2150
                               03  12 0002C          BNEQ    2$                                     ;
                52             07  CE 0002E          MNEGL   #7, LUN_NO                             ;
                               66  D5 00031 2$:      TSTL    SYM$Q_ROOT                             ; 2156
                               1D  12 00033          BNEQ    4$                                     ;
                               7E  D4 00035          CLRL    -(SP)                                  ; 2163
                68             01  FB 00037          CALLS   #1, SYS$SETAST                         ;
                               66  D5 0003A          TSTL    SYM$Q_ROOT                             ; 2165
                               0A  12 0003C          BNEQ    3$                                     ;
                51             66  9E 0003E          MOVAB   SYM$Q_ROOT, R1                         ; 2168
          04    A6             51  D0 00041          MOVL    R1, SYM$Q_ROOT+4                       ;
                66             51  D0 00045          MOVL    R1, SYM$Q_ROOT                         ;
                09             50  D1 00048 3$:      CMPL    AST_STATUS, #9                         ; 2171
                               05  12 0004B          BNEQ    4$                                     ;
                               01  DD 0004D          PUSHL   #1                                     ;
                68             01  FB 0004F          CALLS   #1, SYS$SETAST                         ;
                50             07  CE 00052 4$:      MNEGL   #7, R0                                 ; 2178
                   00000000G   00  16 00055          JSB     BAS$$CB_PUSH                           ;
          FF4C  CB       0C    A3  D0 0005B          MOVL    12(FMP), -180(CCB)                     ; 2179
                54       EC    AB  D0 00061          MOVL    -20(CCB), RBF                          ; 2180
                55       D2    AB  3C 00065          MOVZWL  -46(CCB), RSZ                          ; 2181
                52             66  D0 00069          MOVL    SYM$Q_ROOT, SYM                        ; 2186
                               53  D4 0006C          CLRL    SEARCH_DONE                            ; 2187
                50             66  9E 0006E 5$:      MOVAB   SYM$Q_ROOT, R0                         ; 2192
                50             52  D1 00071          CMPL    SYM, R0                                ;
                               05  12 00074          BNEQ    6$                                     ;
                53             01  D0 00076          MOVL    #1, SEARCH_DONE                        ; 2194
                               56  11 00079          BRB     10$                                    ;
             04    AC       08    A2  D1 0007B 6$:      CMPL    8(SYM), CHAN                       ; 2197
                               4C  12 00080          BNEQ    9$                                     ;
          50 0C    A2       10    A2  C1 00082          ADDL3   16(SYM), 12(SYM), R0               ; 2201
                55             50  D1 00088          CMPL    R0, RSZ                                ;
                               10  14 0008B          BGTR    7$                                     ;
                50       18    A2  D0 0008D          MOVL    24(SYM), VAR                           ; 2212
             04    A0       0C B244  9E 00091          MOVAB   @12(SYM)[RBF], 4(VAR)                  ; 2213
             1C    A2             01  8A 00097          BICB2   #1, 28(SYM)                            ; 2217
                               31  11 0009B          BRB     9$                                     ; 2218
             04    AF             62  0F 0009D 7$:      REMQUE  (SYM), TEMP                        ; 2234
                   50       18    A2  D0 000A1          MOVL    24(SYM), VAR                       ; 2235
                               60  B4 000A5          CLRW    (VAR)                                  ; 2236
             03    A0             02  90 000A7          MOVB    #2, 3(VAR)                         ; 2237
                            04    A0  D4 000AB          CLRL    4(VAR)                             ; 2238
                            04    AE  9F 000AE          PUSHAB  TEMP                               ; 2239
             04    AE             20  D0 000B1          MOVL    #32, 4(SP)                         ;
                            04    AE  9F 000B5          PUSHAB  4(SP)                              ;
       00000000G   00             02  FB 000B8          CALLS   #2, LIB$FREE_VM                    ;
                   07             50  E8 000BF          BLBS    FREE_VM_STATUS, 8$                 ; 2241
                   7E       00G   8F  9A 000C2          MOVZBL  #BAS$K_PROLOSSOR, -(SP)            ;
```

```
                67              01  FB 000C6          CALLS   #1, BAS$$STOP                           ;
                52              66  D0 000C9 8$:       MOVL    SYM$Q_ROOT, SYM                        ; 2243
                                03  11 000CC          BRB     10$                                    ; 2199
                52              62  D0 000CE 9$:       MOVL    (SYM), SYM                             ; 2248
                9A              53  E9 000D1 10$:      BLBC    SEARCH_DONE, 5$                        ; 2251
                    00000000G   00  16 000D4          JSB     BAS$$CB_POP                            ; 2256
                                    04 000DA          RET                                            ; 2257

; Routine Size:  219 bytes,    Routine Base:  _BAS$CODE + 04A1

; 1237          2258  1
```

```
;  1239          2259  1 GLOBAL ROUTINE BAS$FIELD_CLOSE (                  ! Account for CLOSEing a file
;  1240          2260  1         CHAN                                      ! Channel about to be CLOSEed
;  1241          2261  1     ) : NOVALUE =
;  1242          2262  1
;  1243          2263  1 !++
;  1244          2264  1 ! FUNCTIONAL DESCRIPTION:
;  1245          2265  1 !
;  1246          2266  1 !       Account for CLOSEing a file.  Unfield all of the variables
;  1247          2267  1 !       on this channel.
;  1248          2268  1 !
;  1249          2269  1 ! FORMAL PARAMETERS:
;  1250          2270  1 !
;  1251          2271  1 !       CHAN.rl.v       The channel number of the file about to be
;  1252          2272  1 !                       CLOSEed.
;  1253          2273  1 !
;  1254          2274  1 ! IMPLICIT INPUTS:
;  1255          2275  1 !
;  1256          2276  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  1257          2277  1 !
;  1258          2278  1 ! IMPLICIT OUTPUTS:
;  1259          2279  1 !
;  1260          2280  1 !       SYM$Q_ROOT.mq
;  1261          2281  1 !       LUB$V_FIELD_USE for this channel, set to 0
;  1262          2282  1 !
;  1263          2283  1 ! ROUTINE VALUE:
;  1264          2284  1 ! COMPLETION CODES:
;  1265          2285  1 !
;  1266          2286  1 !       NONE
;  1267          2287  1 !
;  1268          2288  1 ! SIDE EFFECTS:
;  1269          2289  1 !
;  1270          2290  1 !       May remove symbols from the symbol table.
;  1271          2291  1 !
;  1272          2292  1 !--
;  1273          2293  1
;  1274          2294  2     BEGIN
;  1275          2295  2
;  1276          2296  2     GLOBAL REGISTER
;  1277          2297  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  1278          2298  2
;  1279          2299  2     LOCAL
;  1280          2300  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  1281          2301  2         LUN_NO,
;  1282          2302  2         SEARCH_DONE;
;  1283          2303  2
;  1284          2304  2 !+
;  1285          2305  2 ! If the symbol table root has not yet been initialized, initialize it.
;  1286          2306  2 !-
;  1287          2307  2
;  1288          2308  3     IF (.SYM$Q_ROOT [0] EQL 0)
;  1289          2309  2     THEN
;  1290          2310  3         BEGIN
;  1291          2311  3
;  1292          2312  3         LOCAL
;  1293          2313  3             AST_STATUS;
;  1294          2314  3
;  1295          2315  3         AST_STATUS = $SETAST (ENBFLG = 0);
```

```
;  1296          2316  3
;  1297          2317  4          IF (.SYM$Q_ROOT [0] EQL 0)
;  1298          2318  3          THEN
;  1299          2319  4              BEGIN
;  1300          2320  4              SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  1301          2321  3              END;
;  1302          2322  3
;  1303          2323  3          IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  1304          2324  3
;  1305          2325  2          END;
;  1306          2326  2
;  1307          2327  2      !+
;  1308          2328  2      ! Search the queue, removing any variables for this channel.
;  1309          2329  2      !-
;  1310          2330  2          SYM = .SYM$Q_ROOT [0];
;  1311          2331  2          SEARCH_DONE = 0;
;  1312          2332  2
;  1313          2333  2          DO
;  1314          2334  3              BEGIN
;  1315          2335  3
;  1316          2336  4              IF (.SYM EQLA SYM$Q_ROOT)
;  1317          2337  3              THEN
;  1318          2338  3                  SEARCH_DONE = 1
;  1319          2339  3              ELSE
;  1320          2340  3
;  1321          2341  4                  IF (.SYM [SYM$L_CHAN] EQL .CHAN)
;  1322          2342  3                  THEN
;  1323          2343  4                      BEGIN
;  1324          2344  4      !+
;  1325          2345  4      ! We must delete this symbol from the symbol table.
;  1326          2346  4      !-
;  1327          2347  4
;  1328          2348  4                      BUILTIN
;  1329          2349  4                          REMQUE;
;  1330          2350  4
;  1331          2351  4                      LOCAL
;  1332          2352  4                          FREE_VM_STATUS,
;  1333          2353  4                          TEMP,
;  1334          2354  4                          VAR : REF BLOCK [8, BYTE];
;  1335          2355  4
;  1336          2356  4                      REMQUE (.SYM, TEMP);
;  1337          2357  4
;  1338          2358  4                      IF (.SYM [SYM$V_INVALID]) THEN BAS$$STOP (BAS$K_ILLFIEVAR);
;  1339          2359  4
;  1340          2360  4                      VAR = .SYM [SYM$A_VAR];
;  1341          2361  4                      VAR [DSC$W_LENGTH] = 0;
;  1342          2362  4                      VAR [DSC$B_CLASS] = DSC$K_CLASS_D;
;  1343          2363  4                      VAR [DSC$A_POINTER] = 0;
;  1344          2364  4                      FREE_VM_STATUS = LIB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
;  1345          2365  4
;  1346          2366  4                      IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
;  1347          2367  4
;  1348          2368  4                      SYM = .SYM$Q_ROOT [0];
;  1349          2369  4                      END
;  1350          2370  3                  ELSE
;  1351          2371  3                      SYM = .SYM [SYM$A_NEXT];
;  1352          2372  3
```

```
;  1353          2373  3          END
;  1354          2374  2      UNTIL (.SEARCH_DONE);
;  1355          2375  2
;  1356          2376  2  !+
;  1357          2377  2  ! load register CCB.
;  1358          2378  2  !-
;  1359          2379  2      LUN_NO = (IF (.CHAN EQL 0) THEN LUB$K_LUN_INPU ELSE .CHAN);
;  1360          2380  2      BAS$$CB_PUSH (.LUN_NO, LUB$K_ILUN_MIN);
;  1361          2381  2
;  1362          2382  2  !+
;  1363          2383  2  ! indicate there is not FIELDing on this channel anymore.
;  1364          2384  2  !-
;  1365          2385  2      CCB [LUB$V_FIELD_USE] = 0;
;  1366          2386  2
;  1367          2387  2  !+
;  1368          2388  2  ! done with register CCB.
;  1369          2389  2  !-
;  1370          2390  2      BAS$$CB_POP ();
;   371          2391  2
;  1372          2392  1      END;                                    ! end of BAS$FIELD_CLOSE


                                 083C 00000          .ENTRY  BAS$FIELD_CLOSE, Save R2,R3,R4,R5,R11       ; 2259
55 00000000G   00    9E 00002    MOVAB   BAS$$STOP, R5                                                   ;
54 00000000G   00    9E 00009    MOVAB   SYS$SETAST, R4                                                  ;
53 00000000'   EF    9E 00010    MOVAB   SYM$Q_ROOT, R3                                                  ;
5E             08    C2 00017    SUBL2   #8, SP                                                          ;
               63    D5 0001A    TSTL    SYM$Q_ROOT                                                      ; 2308
               1D    12 0001C    BNEQ    2$                                                              ;
               7E    D4 0001E    CLRL    -(SP)                                                           ; 2315
         64          01    FB 00020    CALLS   #1, SYS$SETAST                                          ;
               63    D5 00023    TSTL    SYM$Q_ROOT                                                      ; 2317
               0A    12 00025    BNEQ    1$                                                              ;
         51          63    9E 00027    MOVAB   SYM$Q_ROOT, R1                                          ; 2320
04       A3          51    D0 0002A    MOVL    R1, SYM$Q_ROOT+4                                        ;
         63          51    D0 0002E    MOVL    R1, SYM$Q_ROOT                                          ;
         09          50    D1 00031 1$:    CMPL    AST_STATUS, #9                                      ; 2323
               05    12 00034    BNEQ    2$                                                              ;
               01    DD 00036    PUSHL   #1                                                              ;
         64          01    FB 00038    CALLS   #1, SYS$SETAST                                          ;
         52          63    D0 0003B 2$:    MOVL    SYM$Q_ROOT, SYM                                     ; 2330
               5B    D4 0003E    CLRL    SEARCH_DONE                                                     ; 2331
         50          63    9E 00040 3$:    MOVAB   SYM$Q_ROOT, R0                                      ; 2336
         50          52    D1 00043    CMPL    SYM, R0                                                 ;
               05    12 00046    BNEQ    4$                                                              ;
         5B          01    D0 00048    MOVL    #1, SEARCH_DONE                                         ; 2338
               46    11 0004B    BRB     8$                                                              ;
04       AC    08    A2    D1 0004D 4$:    CMPL    8(SYM), CHAN                                        ; 2341
               3C    12 00052    BNEQ    7$                                                              ;
04       AE          62    0F 00054    REMQUE  (SYM), TEMP                                             ; 2356
         07    1C    A2    E9 00058    BLBC    28(SYM), 5$                                             ; 2358
         7E    00G   8F    9A 0005C    MOVZBL  #BAS$K_ILLFIEVAR, -(SP)                                 ;
         65          01    FB 00060    CALLS   #1, BAS$$STOP                                           ;
         50    18    A2    D0 00063 5$:    MOVL    24(SYM), VAR                                        ; 2360
```

```
                          6C  B4 00067          CLRW    (VAR)                                   ; 2361
                    03    02  90 00069          MOVB    #2, 3(VAR)                              ; 2362
                       04 A0  D4 0006D          CLRL    4(VAR)                                  ; 2363
                       04 AE  9F 00070          PUSHAB  TEMP                                    ; 2364
                    04    AE  20  D0 00073          MOVL    #32, 4(SP)                         ;
                       04 AE  9F 00077          PUSHAB  4(SP)                                   ;
              00000000G  00    02  FB 0007A          CALLS   #2, LIB$FREE_VM                   ;
                         07    50  E8 00081          BLBS    FREE_VM_STATUS, 6$                ; 2366
                         7E 00G  8F  9A 00084          MOVZBL  #BAS$K_PROLOSSOR, -(SP)         ;
                         65    01  FB 00088          CALLS   #1, BAS$$STOP                     ;
                         52    63  D0 0008B 6$:       MOVL    SYM$Q_ROOT, SYM                   ; 2368
                               03  11 0008E          BRB     8$                                ; 2341
                         52    62  D0 00090 7$:       MOVL    (SYM), SYM                        ; 2371
                         AA    5B  E9 00093 8$:       BLBC    SEARCH_DONE, 3$                   ; 2374
                       04 AC   D5  00096          TSTL    CHAN                                    ; 2379
                               05  12 00099          BNEQ    9$                                ;
                         52    07  CE 0009B          MNEGL   #7, LUN_NO                        ;
                               04  11 0009E          BRB     10$                               ;
                         52 04 AC  D0 000A0 9$:       MOVL    CHAN, LUN_NO                      ;
                         50    08  CE 000A4 10$:      MNEGL   #8, R0                            ; 2380
                  00000000G    00  16 000A7          JSB     BAS$$CB_PUSH                      ;
                    A1 AB   40    8F  8A 000AD          BICB2   #64, -95(CCB)                   ; 2385
                  00000000G    00  16 000B2          JSB     BAS$$CB_POP                       ; 2390
                               04  000B8          RET                                       ; 2392

; Routine Size:  185 bytes,    Routine Base:  _BAS$CODE + 057C

; 1373          2393  1
```

```
;  1375          2394  1 ROUTINE BAS$$FIELD_KILL                       ! CLOSE appendage
;  1376          2395  1     : CALL_CCB NOVALUE =
;  1377          2396  1
;  1378          2397  1 !++
;  1379          2398  1 ! FUNCTIONAL DESCRIPTION:
;  1380          2399  1 !
;  1381          2400  1 !       This routine is called while a file is being CLOSEd, for any
;  1382          2401  1 !       reason.  If the CLOSE was explicit and in the module containing
;  1383          2402  1 !       the FIELD statement(s), BAS$FIELD_CLOSE will already have
;  1384          2403  1 !       removed all of the field variables for this channel from the
;  1385          2404  1 !       symbol table, so this routine will find none.  If the CLOSE
;  1386          2405  1 !       is implicit or outside the module with the FIELD statement(s),
;  1387          2406  1 !       BAS$FIELD_CLOSE will not have been called and this routine
;  1388          2407  1 !       will mark some variables invalid.  An explicit CLOSE from
;  1389          2408  1 !       another module is considered a programming error, so it is
;  1390          2409  1 !       proper to give an error as soon as any of these variables are
;  1391          2410  1 !       referenced.  We cannot signal an error from here because this
;  1392          2411  1 !       may be the CLOSE from the exit handler (in which case the
;  1393          2412    !       variables will not be referenced again, so marking them
;  1394          2413    !       invalid is OK) or the implicit CLOSE from OPEN, in which case
;  1395          2414  1 !       (if the OPEN is from a module with FIELD) BAS$FIELD_OPEN will
;  1396          2415  1 !       re-validate the variables still in the buffer.
;  1397          2416  1 !
;  1398          2417  1 ! FORMAL PARAMETERS:
;  1399          2418  1 !
;  1400          2419  1 !       NONE
;  1401          2420  1 !
;  1402          2421  1 ! IMPLICIT INPUTS:
;  1403          2422  1 !
;  1404          2423  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  1405          2424  1 !       LUB$W_LUN       The logical unit number of the file being closed
;  1406          2425  1 !
;  1407          2426  1 ! IMPLICIT OUTPUTS:
;  1408          2427  1 !
;  1409          2428  1 !       SYM$Q_ROOT.mq
;  1410          2429  1 !
;  1411          2430  1 ! ROUTINE VALUE:
;  1412          2431  1 ! COMPLETION CODES:
;  1413          2432  1 !
;  1414          2433  1 !       NONE
;  1415          2434  1 !
;  1416          2435  1 ! SIDE EFFECTS:
;  1417          2436  1 !
;  1418          2437  1 !       May mark symbols invalid, but is most likely to have no net
;  1419          2438  1 !       effect.
;  1420          2439  1 !
;  1421          2440  1 !--
;  1422          2441  1
;  1423          2442  2     BEGIN
;  1424          2443  2
;  1425          2444  2     EXTERNAL REGISTER
;  1426          2445  2         CCB : REF BLOCK [, BYTE];
;  1427          2446  2
;  1428          2447  2     LOCAL
;  1429          2448  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  1430          2449  2         SEARCH_DONE,
;  1431          2450  2         CHAN;
```

```
;  1432     2451  2
;  1433     2452  2 !+
;  1434     2453  2 ! If the symbol table root has not yet been initialized, initialize it.
;  1435     2454  2 !-
;  1436     2455  2
;  1437     2456  3     IF (.SYMSQ_ROOT [0] EQL 0)
;  1438     2457  3     THEN
;  1439     2458  3         BEGIN
;  1440     2459  3
;  1441     2460  3         LOCAL
;  1442     2461  3             AST_STATUS;
;  1443     2462  3
;  1444     2463  3         AST_STATUS = $SETAST (ENBFLG = 0);
;  1445     2464  3
;  1446     2465  4         IF (.SYMSQ_ROOT [0] EQL 0)
;  1447     2466  3         THEN
;  1448     2467  4             BEGIN
;  1449     2468  4             SYMSQ_ROOT [0] = SYMSQ_ROOT [1] = SYMSQ_ROOT [0];
;  1450     2469  3             END;
;  1451     2470  3
;  1452     2471  3         IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  1453     2472  3
;  1454     2473  2         END;
;  1455     2474  2
;  1456     2475  2 !+
;  1457     2476  2 ! Compute the channel number from the logical unit number.
;  1458     2477  2 !-
;  1459     2478  2     CHAN = (IF (.CCB [LUB$W_LUN] EQL LUB$K_LUN_INPU) THEN 0 ELSE .CCB [LUB$W_LUN]);
;  1460     2479  2 !+
;  1461     2480  2 ! Search the queue, invalidating any variables for this channel.
;  1462     2481  2 !-
;  1463     2482  2     SYM = .SYMSQ_ROOT [0];
;  1464     2483  2     SEARCH_DONE = 0;
;  1465     2484  2
;  1466     2485  2     DO
;  1467     2486  3         BEGIN
;  1468     2487  3
;  1469     2488  4         IF (.SYM EQLA SYMSQ_ROOT)
;  1470     2489  3         THEN
;  1471     2490  3             SEARCH_DONE = 1
;  1472     2491  3         ELSE
;  1473     2492  3
;  1474     2493  4             IF (.SYM [SYM$L_CHAN] EQL .CHAN)
;  1475     2494  3             THEN
;  1476     2495  4                 BEGIN
;  1477     2496  4 !+
;  1478     2497  4 ! We must mark this symbol as invalid.
;  1479     2498  4 !-
;  1480     2499  4
;  1481     2500  4                 LOCAL
;  1482     2501  4                     VAR : REF BLOCK [8, BYTE];
;  1483     2502  4
;  1484     2503  4                 VAR = .SYM [SYM$A_VAR];
;  1485     2504  4                 VAR [DSC$A_POINTER] = 0;
;  1486     2505  4                 SYM [SYM$V_INVALID] = 1;
;  1487     2506  3                 END;
;  1488     2507  3
```

```
;  1489          2508  3          SYM = .SYM [SYM$A_NEXT];
;  1490          2509  3          END
;  1491          2510  2      UNTIL (.SEARCH_DONE);
;  1492          2511  2
;  1493          2512  1      END;                                    end of BAS$$FIELD_KILL


                                        003C 00000 BAS$$FIELD_KILL:
                                                   .WORD   Save R2,R3,R4,R5                    ; 2394
                    55 00000000G    00  9E 00002           MOVAB   SYS$SETAST, R5
                    54 00000000'    EF  9E 00009           MOVAB   SYM$Q_ROOT, R4
                                    64  D5 00010           TSTL    SYM$Q_ROOT                  ; 2456
                                    1D  12 00012           BNEQ    2$
                                    7E  D4 00014           CLRL    -(SP)                       ; 2463
                    65              01  FB 00016           CALLS   #1, SYS$SETAST
                                    64  D5 00019           TSTL    SYM$Q_ROOT                  ; 2465
                                    0A  12 0001B           BNEQ    1$
                    51              64  9E 0001D           MOVAB   SYM$Q_ROOT, R1              ; 2468
                04  A4              51  D0 00020           MOVL    R1, SYM$Q_ROOT+4
                    64              51  D0 00024           MOVL    R1, SYM$Q_ROOT
                    09              50  D1 00027 1$:       CMPL    AST_STATUS, #9              ; 2471
                                    05  12 0002A           BNEQ    2$
                                    01  DD 0002C           PUSHL   #1
                    65              01  FB 0002E           CALLS   #1, SYS$SETAST
                FFF9  8F        C6  AB  B1 00031 2$:       CMPW    -58(CCB), #-7               ; 2478
                                    04  12 00037           BNEQ    3$
                                    52  D4 00039           CLRL    CHAN
                                    04  11 0003B           BRB     4$
                        52      C6  AB  32 0003D 3$:       CVTWL   -58(CCB), CHAN
                        51          64  D0 00041 4$:       MOVL    SYM$Q_ROOT, SYM             ; 2482
                                    53  D4 00044           CLRL    SEARCH_DONE                 ; 2483
                        50          64  9E 00046 5$:       MOVAB   SYM$Q_ROOT, R0              ; 2488
                        50          51  D1 00049           CMPL    SYM, R0
                                    05  12 0004C           BNEQ    6$
                        53          01  D0 0004E           MOVL    #1, SEARCH_DONE             ; 2490
                                    11  11 00051           BRB     7$
                        52      08  A1  D1 00053 6$:       CMPL    8(SYM), CHAN                ; 2493
                                    0B  12 00057           BNEQ    7$
                        50      18  A1  D0 00059           MOVL    24(SYM), VAR                ; 2503
                                04  A0  D4 0005D           CLRL    4(VAR)                      ; 2504
                  1C    A1          01  88 00060           BISB2   #1, 28(SYM)                 ; 2505
                        51          61  D0 00064 7$:       MOVL    (SYM), SYM                  ; 2508
                        DC          53  E9 00067           BLBC    SEARCH_DONE, 5$             ; 2510
                                        04 0006A           RET                                 ; 2512

; Routine Size:  107 bytes,    Routine Base:  _BAS$CODE + 0635

;  1494          2513  1
```

```
;  1496          2514  1 GLOBAL ROUTINE BAS$$FIELD_INIT : NOVALUE =      ! Initialize for RUN
;  1497          2515  1
;  1498          2516  1 !++
;  1499          2517  1 ! FUNCTIONAL DESCRIPTION:
;  1500          2518  1 !
;  1501          2519  1 !       Initialize the FIELD symbol table for the RUN command.  All symbols are removed
;  1502          2520  1 !       from the table, even those marked invalid.
;  1503          2521  1 !
;  1504          2522  1 ! FORMAL PARAMETERS:
;  1505          2523  1 !
;  1506          2524  1 !       NONE
;  1507          2525  1 !
;  1508          2526  1 ! IMPLICIT INPUTS:
;  1509          2527  1 !
;  1510          2528  1 !       SYM$Q_ROOT.mq   The queue of FIELD variables : the symbol table.
;  1511          2529  1 !
;  1512          2530  1 ! IMPLICIT OUTPUTS:
;  1513          2531  1 !
;  1514          2532  1 !       SYM$Q_ROOT.mq
;  1515          2533  1 !
;  1516          2534  1 ! ROUTINE VALUE:
;  1517          2535  1 ! COMPLETION CODES:
;  1518          2536  1 !
;  1519          2537  1 !       NONE
;  1520          2538  1 !
;  1521          2539  1 ! SIDE EFFECTS:
;  1522          2540  1 !
;  1523          2541  1 !       Makes the symbol table empty.
;  1524          2542  1 !
;  1525          2543  1 !--
;  1526          2544  1
;  1527          2545  2     BEGIN
;  1528          2546  2
;  1529          2547  2     LOCAL
;  1530          2548  2         SYM : REF BLOCK [SYM$K_LENGTH, BYTE] FIELD (BAS$FIELD_SYM),
;  1531          2549  2         SEARCH_DONE;
;  1532          2550  2
;  1533          2551  2 !+
;  1534          2552  2 ! If the symbol table root has not yet been initialized, initialize it.
;  1535          2553  2 !-
;  1536          2554  2
;  1537          2555  3     IF (.SYM$Q_ROOT [0] EQL 0)
;  1538          2556  2     THEN
;  1539          2557  3         BEGIN
;  1540          2558  3
;  1541          2559  3         LOCAL
;  1542          2560  3             AST_STATUS;
;  1543          2561  3
;  1544          2562  3         AST_STATUS = $SETAST (ENBFLG = 0);
;  1545          2563  3
;  1546          2564  4         IF (.SYM$Q_ROOT [0] EQL 0)
;  1547          2565  3         THEN
;  1548          2566  4             BEGIN
;  1549          2567  4             SYM$Q_ROOT [0] = SYM$Q_ROOT [1] = SYM$Q_ROOT [0];
;  1550          2568  3             END;
;  1551          2569  3
;  1552          2570  3         IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
```

```
;  1553          2571  3
;  1554          2572  2          END;
;  1555          2573  2
;  1556          2574  2  !+
;  1557          2575  2  ! Search the queue, deleting any symbols in it.
;  1558          2576  2  !-
;  1559          2577  2      SYM = .SYM$Q_ROOT [0];
;  1560          2578  2      SEARCH_DONE = 0;
;  1561          2579  2
;  1562          2580  2      DO
;  1563          2581  3          BEGIN
;  1564          2582  3
;  1565          2583  4          IF (.SYM EQLA SYM$Q_ROOT)
;  1566          2584  3          THEN
;  1567          2585  3              SEARCH_DONE = 1
;  1568          2586  3          ELSE
;  1569          2587  4              BEGIN
;  1570          2588  4  !+
;  1571          2589  4  ! We must delete this symbol from the symbol table.
;  1572          2590  4  !-
;  1573          2591  4
;  1574          2592  4              BUILTIN
;  1575          2593  4                  REMQUE;
;  1576          2594  4
;  1577          2595  4              LOCAL
;  1578          2596  4                  FREE_VM_STATUS,
;  1579          2597  4                  TEMP,
;  1580          2598  4                  VAR : REF BLOCK [8, BYTE];
;  1581          2599  4
;  1582          2600  4              REMQUE (.SYM, TEMP);
;  1583          2601  4              VAR = .SYM [SYM$A_VAR];
;  1584          2602  4              FREE_VM_STATUS = $IB$FREE_VM (%REF (SYM$K_LENGTH), TEMP);
;  1585          2603  4
;  1586          2604  4              IF ( NOT .FREE_VM_STATUS) THEN BAS$$STOP (BAS$K_PROLOSSOR);
;  1587          2605  4
;  1588          2606  4              SYM = .SYM$Q_ROOT [0];
;  1589          2607  4              END
;  1590          2608  4
;  1591          2609  3          END
;  1592          2610  2      UNTIL (.SEARCH_DONE);
;  1593          2611  2
;  1594          2612  1      END;                                    ! end of BAS$$FIELD_INIT


                                    003C 00000          .ENTRY  BAS$$FIELD_INIT, Save R2,R3,R4,R5       ; 2514
                55 00000000G    00  9E 00002          MOVAB   SYS$SETAST, R5                          ;
                54 00000000'    EF  9E 00009          MOVAB   SYM$Q_ROOT, R4                          ;
                5E                  08  C2 00010      SUBL2   #8, SP                                  ;
                                    64  D5 00013      TSTL    SYM$Q_ROOT                              ; 2555
                                    1D  12 00015      BNEQ    2$                                      ;
                                    7E  D4 00017      CLRL    -(SP)                                   ; 2562
                65                  01  FB 00019      CALLS   #1, SYS$SETAST                          ;
                                    64  D5 0001C      TSTL    SYM$Q_ROOT                              ; 2564
                                    0A  12 0001E      BNEQ    1$                                      ;
```

```
                        51              64  9E 00020         MOVAB   SYM$Q_ROOT, R1                    ; 2567
                    04  A4              51  D0 00023         MOVL    R1, SYM$Q_ROOT+4                 ;
                        64              51  D0 00027         MOVL    R1, SYM$Q_ROOT                   ;
                        09              50  D1 0002A 1$:      CMPL    AST_STATUS, #9                   ; 2570
                                        05  12 0002D         BNEQ    2$                               ;
                                        01  DD 0002F         PUSHL   #1                               ;
                        65              01  FB 00031         CALLS   #1, SYS$SETAST                   ;
                        52              64  D0 00034 2$:      MOVL    SYM$Q_ROOT, SYM                  ; 2577
                                        53  D4 00037         CLRL    SEARCH_DONE                      ; 2578
                        50              64  9E 00039 3$:      MOVAB   SYM$Q_ROOT, R0                   ; 2583
                        50              52  D1 0003C         CMPL    SYM, R0                          ;
                                        05  12 0003F         BNEQ    4$                               ;
                        53              01  D0 00041         MOVL    #1, SEARCH_DONE                  ; 2585
                                        2A  11 00044         BRB     6$                               ;
                    04  AE              62  0F 00046 4$:      REMQUE  (SYM), TEMP                      ; 2600
                        50          18  A2  D0 0004A         MOVL    24(SYM), VAR                     ; 2601
                                    04  AE  9F 0004E         PUSHAB  TEMP                             ; 2602
                    04  AE              20  D0 00051         MOVL    #32, 4(SP)                       ;
                                    04  AE  9F 00055         PUSHAB  4(SP)                            ;
                00000000G   00          02  FB 00058         CALLS   #2, LIB$FREE_VM                  ;
                            0B          50  E8 0005F         BLBS    FREE_VM_STATUS, 5$               ; 2604
                            7E      00G 8F  9A 00062         MOVZBL  #BAS$K_PROLOSSOR, -(SP)          ;
                00000000G   00          01  FB 00066         CALLS   #1, BAS$$STOP                    ;
                            52          64  D0 0006D 5$:      MOVL    SYM$Q_ROOT, SYM                  ; 2606
                            C6          53  E9 00070 6$:      BLBC    SEARCH_DONE, 3$                  ; 2610
                                            04 00073         RET                                      ; 2612

; Routine Size:  116 bytes,    Routine Base:  _BAS$CODE + 06A0

; 1595          2613  1
; 1596          2614  1 END                                     ! end of module BAS$RSTS_FIELD
; 1597          2615  1
; 1598          2616  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
_BAS$DATA	8	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON, PIC,ALIGN(2)
_BAS$CODE	1812	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time
_255DUA28:[SYSLIB]STARLET.L32;1	9776	12	0	581	00:01.2

```
;                                       COMMAND QUALIFIERS

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

; Size:         1812 code + 8 data bytes
; Run Time:        00:40.2
; Elapsed Time:    01:22.3
; Lines/CPU Min:    3900
; Lexemes/CPU-Min: 25164
; Memory Used:  227 pages
; Compilation Complete
```