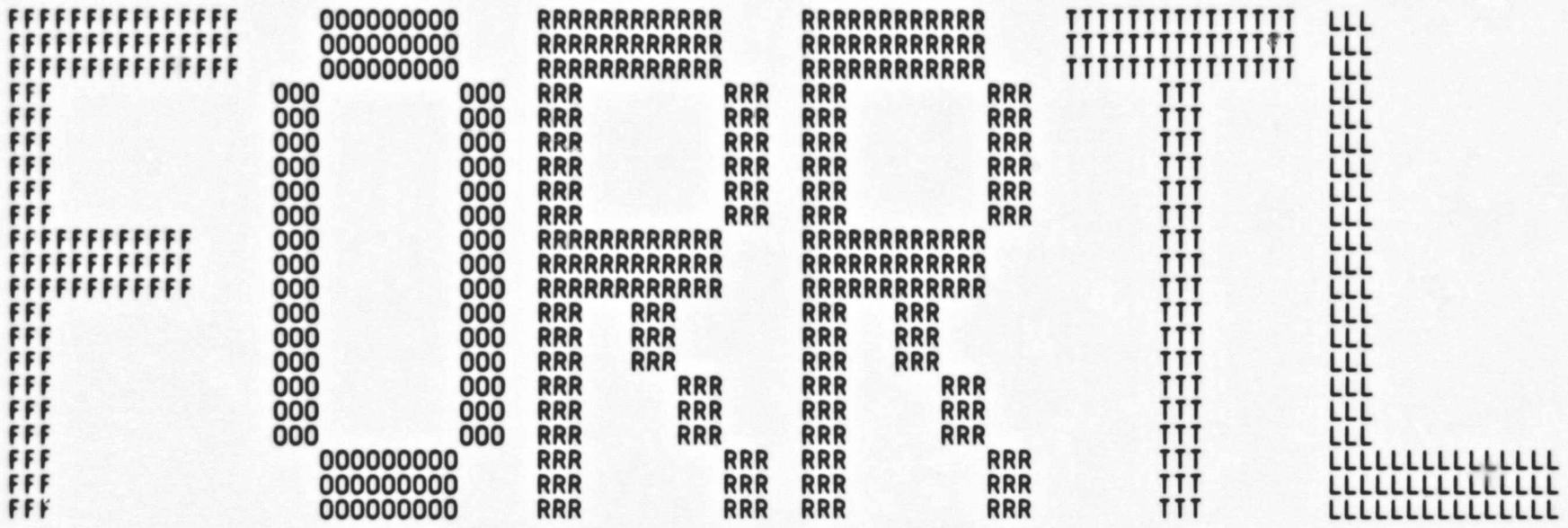
FORTRAN
LLLLLLLLLLLLLLL

FILEID**FORINQUIR

```
FFFFFFFFFF    000000    RRRRRRRR     IIIIII    NN      NN    QQQQQQ    UU      UU    IIIIII    RRRRRRRR
FFFFFFFFFF    000000    RRRRRRRR     IIIIII    NN      NN    QQQQQQ    UU      UU    IIIIII    RRRRRRRR
FF          00      00  RR      RR     II      NN      NN  QQ      QQ  UU      UU      II      RR      RR
FF          00      00  RR      RR     II      NN      NN  QQ      QQ  UU      UU      II      RR      RR
FF          00      00  RR      RR     II      NNNN    NN  QQ      QQ  UU      UU      II      RR      RR
FF          00      00  RR      RR     II      NNNN    NN  QQ      QQ  UU      UU      II      RR      RR
FFFFFFFF    00      00  RRRRRRRR       II      NN  NN  NN  QQ      QQ  UU      UU      II      RRRRRRRR
FFFFFFFF    00      00  RRRRRRRR       II      NN  NN  NN  QQ      QQ  UU      UU      II      RRRRRRRR
FF          00      00  RR  RR         II      NN    NNNN  QQ  QQ  QQ  UU      UU      II      RR  RR
FF          00      00  RR  RR         II      NN    NNNN  QQ  QQ  QQ  UU      UU      II      RR  RR
FF          00      00  RR    RR       II      NN      NN  QQ    QQ    UU      UU      II      RR    RR
FF          00      00  RR    RR       II      NN      NN  QQ    QQ    UU      UU      II      RR    RR
FF            000000    RR      RR   IIIIII    NN      NN    QQQQ  QQ  UUUUUUUUUU    IIIIII    RR      RR
FF            000000    RR      RR   IIIIII    NN      NN    QQQQ  QQ  UUUUUUUUUU    IIIIII    RR      RR

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

```
;     1          0001  0 MODULE FOR$INQUIRE (%TITLE'FORTRAN INQUIRE'
;     2          0002  0                     IDENT = '1-017'          ! File: FORINQUIR.B32  Edit: SBL1017
;     3          0003  0                     ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !*****************************************************************************
;     8          0008  1 !*                                                                           *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                     *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
;    18          0018  1 !*  TRANSFERRED.                                                             *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
;    22          0022  1 !*  CORPORATION.                                                             *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY: FORTRAN Language Support Library
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       Implements the FORTRAN INQUIRE statement.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT: User mode, AST reentrant.
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 28-August-1979
;    41          0041  1 !
;    42          0042  1 ! EDIT HISTORY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  SBL 28-August-1979
;    45          0045  1 ! 1-002 - Add comments about AST disabling.  Also report errors
;    46          0046  1 !         RMS$_TYP and RMS$_VER as file name syntax errors.  SBL
;    47          0047  1 ! 1-003 - Instead of giving an error, just pretend units "poisoned"
;    48          0048  1 !         with DEFINE FILE, etc. aren't open.  SBL 21-Sept-1979
;    49          0049  1 ! 1-004 - Improve error handling.  SBL 8-Oct-1979
;    50          0050  1 ! 1-005 - Remove file name editing.  RMS does it now.  SBL 12-Oct-1979
;    51          0051  1 ! 1-006 - If FILEname invalid, return user supplied filename as NAME.  SBL 19-Oct-1979
;    52          0052  1 ! 1-007 - Add CARRIAGECONTROL keyword.  SBL 4-Dec-1979
;    53          0053  1 ! 1-008 - Class RMS$_SYN as file name specification error.  If wildcard
;    54          0054  1 !         present, don't report phony RMS error and don't scan LUNs.  SBL 6-Feb-1980
;    55          0055  1 ! 1-009 - Have NEXTREC simply fetch LUB$L_LOG_RECNO.  Remove the erroneous
;    56          0056  1 !         MIN which caused it to always return 1!  Correct the comment
;    57          0057  1 !         for MAXREC to conform to the code and spec.  Move declaration
```

```
;    58          0058  1 !          of ACTUALCOUNT to the inside of the routine which uses it.
;    59          0059  1 !          SBL 21-August-1980
;    60          0060  1 ! 1-010 - Add support for DEFAULTFILE.  JAW 01-Jul-1981
;    61          0061  1 ! 1-011 - Allow DEFAULTFILE value to be ASCIZ.  JAW 02-Jul-1981
;    62          0062  1 ! 1-012 - Open file for shared access.  SBL 2-Nov-1981
;    63          0063  1 ! 1-013 - Declare LIB$STOP external.  SBL 30-Nov-1981
;    64          0064  1 ! 1-014 - Do extra $PARSE after $SEARCH to make RMS clean up its internal
;    65          0065  1 !          structures.  Never signal any error except PUSH_CB errors in
;    66          0066  1 !          INQUIRE by UNIT.  SPR 11-47083  SBL 5-August-1982
;    67          0067  1 ! 1-015 - Don't do the $SEARCH for non-FOD devices, as it will return
;    68          0068  1 !          RMS$_IOP.  SBL 14-Jan-1983
;    69          0069  1 ! 1-016 - Make use of the fact that the FAB and NAM blocks are heap-allocated.
;    70          0070  1 !          Use prologue file.  SBL 20-Apr-1983
;    71          0071  1 ! 1-017 - Use individual NAM$V_WILD_xxx fields to determine if wildcard is
;    72          0072  1 !          present since search lists would trigger NAM$V_WILDCARD.  SBL 24-Aug-1983
;    73          0073  1 !--
```

```
;    75          0074  1 !
;    76          0075  1 ! PROLOGUE FILE:
;    77          0076  1 !
;    78          0077  1
;    79          0078  1 REQUIRE 'RTLIN:FORPROLOG';              ! FORTRAN-specific definitions
;    80          0144  1
;    81          0145  1 !
;    82          0146  1 ! TABLE OF CONTENTS:
;    83          0147  1 !
;    84          0148  1
;    85          0149  1 FORWARD ROUTINE
;    86          0150  1         FOR$INQUIRE,                    ! Process INQUIRE statement
;    87          0151  1         PUSH_CCB : CALL_CCB;            ! Calls FOR$$CB_PUSH
;    88          0152  1
;    89          0153  1 !
;    90          0154  1 ! FIELDS:
;    91          0155  1 !
;    92          0156  1     FIELD
;    93          0157  1         INQ_FIELDS =
;    94          0158  1             SET
;    95          0159  1             UNIT_OK     = [0,0,1,0],    ! 1 if UNIT is valid
;    96          0160  1             NAME_OK     = [0,1,1,0],    ! 1 if name is valid
;    97          0161  1             CCB_OK      = [0,2,1,0],    ! 1 if CCB valid
;    98          0162  1             EXISTS      = [0,3,1,0],    ! 1 if file exists
;    99          0163  1             FAB_OK      = [0,4,1,0],    ! 1 if FAB is valid
;   100          0164  1             BY_FILE     = [0,5,1,0]     ! 1 if INQUIRE by FILE
;   101          0165  1             TES;
;   102          0166  1
;   103          0167  1 !
;   104          0168  1 ! EQUATED SYMBOLS:
;   105          0169  1 !
;   106          0170  1
;   107          0171  1 !+
;   108          0172  1 ! Codes for string responses.  These values are put into vector
;   109          0173  1 ! RESP_VEC when the correct response is determined.
;   110          0174  1 !-
;   111          0175  1 LITERAL
;   112          0176  1     BLANK                   = 0,
;   113          0177  1     YES                     = 1,
;   114          0178  1     NO                      = 2,
;   115          0179  1     UNKNOWN                 = 3,
;   116          0180  1     DIRECT                  = 4,
;   117          0181  1     KEYED                   = 5,
;   118          0182  1     SEQUENTIAL              = 6,
;   119          0183  1     FORMATTED               = 7,
;   120          0184  1     UNFORMATTED             = 8,
;   121          0185  1     RELATIVE                = 9,
;   122          0186  1     INDEXED                 = 10,
;   123          0187  1     NULL                    = 11,
;   124          0188  1     ZERO                    = 12,
;   125          0189  1     FIXED                   = 13,
;   126          0190  1     VARIABLE                = 14,
;   127          0191  1     SEGMENTED               = 15,
;   128          0192  1     FORTRAN                 = 16,
;   129          0193  1     LIST                    = 17,
;   130          0194  1     NONE                    = 18,
;   131          0195  1     STREAM                  = 19,
```

```
;  132          0196  1     STREAM_CR           = 20,
;  133          0197  1     STREAM_LF           = 21;
;  134          0198  1
;  135          0199  1 !
;  136          0200  1 ! OWN STORAGE:
;  137          0201  1 !
;  138          0202  1
;  139          0203  1 !+
;  140          0204  1 ! String responses are stored here in a PIC table.
;  141          0205  1 !-
;  142          0206  1 OWN
;  143          0207  1     RESP_VALS : VECTOR [22,LONG] PSECT (_FOR$CODE) INITIAL (
;  144          0208  1         UPLIT BYTE (' ') - RESP_VALS,                ! 0
;  145          0209  1         UPLIT BYTE ('YES') - RESP_VALS,              ! 1
;  146          0210  1         UPLIT BYTE ('NO') - RESP_VALS,               ! 2
;  147          0211  1         UPLIT BYTE ('UNKNOWN') - RESP_VALS,          ! 3
;  148          0212  1         UPLIT BYTE ('DIRECT') - RESP_VALS,           ! 4
;  149          0213  1         UPLIT BYTE ('KEYED') - RESP_VALS,            ! 5
;  150          0214  1         UPLIT BYTE ('SEQUENTIAL') - RESP_VALS,       ! 6
;  151          0215  1         UPLIT BYTE ('FORMATTED') - RESP_VALS,        ! 7
;  152          0216  1         UPLIT BYTE ('UNFORMATTED') - RESP_VALS,      ! 8
;  153          0217  1         UPLIT BYTE ('RELATIVE') - RESP_VALS,         ! 9
;  154          0218  1         UPLIT BYTE ('INDEXED') - RESP_VALS,          ! 10
;  155          0219  1         UPLIT BYTE ('NULL') - RESP_VALS,             ! 11
;  156          0220  1         UPLIT BYTE ('ZERO') - RESP_VALS,             ! 12
;  157          0221  1         UPLIT BYTE ('FIXED') - RESP_VALS,            ! 13
;  158          0222  1         UPLIT BYTE ('VARIABLE') - RESP_VALS,         ! 14
;  159          0223  1         UPLIT BYTE ('SEGMENTED') - RESP_VALS,        ! 15
;  160          0224  1         UPLIT BYTE ('FORTRAN') - RESP_VALS,          ! 16
;  161          0225  1         UPLIT BYTE ('LIST') - RESP_VALS,             ! 17
;  162          0226  1         UPLIT BYTE ('NONE') - RESP_VALS,             ! 18
;  163          0227  1         UPLIT BYTE ('STREAM') - RESP_VALS,           ! 19
;  164          0228  1         UPLIT BYTE ('STREAM_CR') - RESP_VALS,        ! 20
;  165          0229  1         UPLIT BYTE ('STREAM_LF') - RESP_VALS         ! 21
;  166          0230  1         );
;  167          0231  1
;  168          0232  1 !+
;  169          0233  1 ! Lengths of string responses.
;  170          0234  1 !-
;  171          0235  1 OWN
;  172          0236  1     RESP_LENS : VECTOR [22,BYTE] PSECT (_FOR$CODE) INITIAL (BYTE (
;  173          0237  1         %CHARCOUNT (' '),                            ! 0
;  174          0238  1         %CHARCOUNT ('YES'),                          ! 1
;  175          0239  1         %CHARCOUNT ('NO'),                           ! 2
;  176          0240  1         %CHARCOUNT ('UNKNOWN'),                      ! 3
;  177          0241  1         %CHARCOUNT ('DIRECT'),                       ! 4
;  178          0242  1         %CHARCOUNT ('KEYED'),                        ! 5
;  179          0243  1         %CHARCOUNT ('SEQUENTIAL'),                   ! 6
;  180          0244  1         %CHARCOUNT ('FORMATTED'),                    ! 7
;  181          0245  1         %CHARCOUNT ('UNFORMATTED'),                  ! 8
;  182          0246  1         %CHARCOUNT ('RELATIVE'),                     ! 9
;  183          0247  1         %CHARCOUNT ('INDEXED'),                      ! 10
;  184          0248  1         %CHARCOUNT ('NULL'),                         ! 11
;  185          0249  1         %CHARCOUNT ('ZERO'),                         ! 12
;  186          0250  1         %CHARCOUNT ('FIXED'),                        ! 13
;  187          0251  1         %CHARCOUNT ('VARIABLE'),                     ! 14
;  188          0252  1         %CHARCOUNT ('SEGMENTED'),                    ! 15
```

```
;  189          0253  1          %CHARCOUNT ('FORTRAN'),          ! 16
;  190          0254  1          %CHARCOUNT ('LIST'),             ! 17
;  191          0255  1          %CHARCOUNT ('NONE'),             ! 18
;  192          0256  1          %CHARCOUNT ('STREAM'),           ! 19
;  193          0257  1          %CHARCOUNT ('STREAM_CR'),        ! 20
;  194          0258  1          %CHARCOUNT ('STREAM_LF')         ! 21
;  195          0259  1          ));
;  196          0260  1
;  197          0261  1  !+
;  198          0262  1  ! Vector which is indexed by the keyword number.  Values are:
;  199          0263  1  ! 0 = do nothing, 1 = numeric, 2 = string.
;  200          0264  1  !-
;  201          0265  1  OWN
;  202          0266  1      RESP_TYPES : VECTOR [INQ$K_KEY_MAX + 1, BYTE] PSECT (_FOR$CODE) INITIAL (BYTE (
;  203          0267  1          REP INQ$K_IOSTAT OF (0),                ! Up to but not including IOSTAT
;  204          0268  1          1,          ! IOSTAT
;  205          0269  1          REP INQ$K_EXIST - (INQ$K_IOSTAT + 1) OF (0),    ! Unused space
;  206          0270  1          1,          ! EXIST
;  207          0271  1          1,          ! OPENED
;  208          0272  1          1,          ! NUMBER
;  209          0273  1          1,          ! NAMED
;  210          0274  1          0,          ! NAME (name is stored separately)
;  211          0275  1          2,          ! ACCESS
;  212          0276  1          2,          ! SEQUENTIAL
;  213          0277  1          2,          ! DIRECT
;  214          0278  1          2,          ! FORM
;  215          0279  1          2,          ! FORMATTED
;  216          0280  1          2,          ! UNFORMATTED
;  217          0281  1          1,          ! RECL
;  218          0282  1          1,          ! NEXTREC
;  219          0283  1          2,          ! BLANK
;  220          0284  1          2,          ! ORGANIZATION
;  221          0285  1          2,          ! RECORDTYPE
;  222          0286  1          2,          ! KEYED
;  223          0287  1          2           ! CARRIAGECONTROL
;  224          0288  1          ));
;  225          0289  1
;  226          0290  1  !
;  227          0291  1  ! EXTERNAL REFERENCES:
;  228          0292  1  !
;  229          0293  1
;  230          0294  1  EXTERNAL ROUTINE
;  231          0295  1      FOR$$ERR_OPECLO,                    ! Error handler
;  232          0296  1      FOR$$ERRSNS_SAV : NOVALUE,          ! Save ERRSNS values
;  233          0297  1      FOR$$OPECLO_ARG,                    ! Process argument list
;  234          0298  1      FOR$$CB_PUSH : JSB_CB_PUSH,         ! Get a CCB
;  235          0299  1      FOR$$CB_POP : JSB_CB_POP NOVALUE,   ! Free a CCB
;  236          0300  1      FOR$$CB_FETCH : CALL_CCB NOVALUE,   ! Fetch a CCB
;  237          0301  1      FOR$$SIG_NO_LUB : NOVALUE,          ! Signal fatal error
;  238          0302  1      LIB$SIG_TO_RET,                     ! Condition handler
;  239          0303  1      LIB$STOP: NOVALUE,                  ! Signal non-continuable error
;  240          0304  1      FOR$$NEXT_LUN : NOVALUE;            ! Find next allocated LUN
```

```
;  242     0305  1 GLOBAL ROUTINE FOR$INQUIRE (
;  243     0306  1     KEYWD) =                                ! Argument list
;  244     0307  1
;  245     0308  1 !++
;  246     0309  1 ! FUNCTIONAL DESCRIPTION:
;  247     0310  1 !
;  248     0311  1 !       Processes the FORTRAN INQUIRE statement.
;  249     0312  1 !
;  250     0313  1 !       There are two ways to inquire - by file and by unit.
;  251     0314  1 !       Inquiring by file returns information about that file
;  252     0315  1 !       and about the unit on which it is opened, if any.
;  253     0316  1 !       Inquiring by unit returns information about that unit
;  254     0317  1 !       and about the file opened on it, if any.
;  255     0318  1 !
;  256     0319  1 ! FORMAL PARAMETERS:
;  257     0320  1 !
;  258     0321  1 !       The entire argument list is comprised of groups of keywords
;  259     0322  1 !       and values.  The format is identical to that for FOR$OPEN.
;  260     0323  1 !
;  261     0324  1 ! IMPLICIT INPUTS:
;  262     0325  1 !
;  263     0326  1 !       NONE
;  264     0327  1 !
;  265     0328  1 ! IMPLICIT OUTPUTS:
;  266     0329  1 !
;  267     0330  1 !       Values are returned to the caller through addresses denoted
;  268     0331  1 !       in the keyword list.
;  269     0332  1 !
;  270     0333  1 ! COMPLETION CODES:
;  271     0334  1 !
;  272     0335  1 !       SS$_NORMAL - Successful completion
;  273     0336  1 !
;  274     0337  1 ! SIDE EFFECTS:
;  275     0338  1 !
;  276     0339  1 !       On INQUIRE by FILE: disables ASTs while looking at a unit
;  277     0340  1 !       to determine if it is open.  Inquiring about a unit or about
;  278     0341  1 !       a file open on a unit is considered I/O on that unit and
;  279     0342  1 !       follows the FORTRAN rules for recursive I/O (i.e., it's
;  280     0343  1 !       not permitted).
;  281     0344  1 !
;  282     0345  1 !--
;  283     0346  1
;  284     0347  2     BEGIN
;  285     0348  2
;  286     0349  2     GLOBAL REGISTER
;  287     0350  2         CCB = 11 : REF $FOR$CCB_DECL;
;  288     0351  2
;  289     0352  2     BUILTIN
;  290     0353  2         ACTUALCOUNT;                        ! Number of arguments in call
;  291     0354  2
;  292     0355  2     MAP
;  293     0356  2         KEYWD : BLOCKVECTOR [255, 1];       ! Use the format arg list
;  294     0357  2
;  295     0358  2     LOCAL
;  296     0359  2         NAM_DSC : DSC$DESCRIPTOR,           ! String descriptor for file name
;  297     0360  2         DEF_DSC : DSC$DESCRIPTOR,           ! String descriptor for default file name
;  298     0361  2         L_UNWIND_ACTION : VOLATILE,         ! UNWIND action code for handler
```

```
;  299          0362  2          INQUIRE : VOLATILE VECTOR [INQ$K_KEY_MAX + 1],  ! INQUIRE parameter array
;  300          0363  2          NAM_BLOCK : $NAM_DECL,                          ! NAM block
;  301          0364  2          FAB_BLOCK : $FAB_DECL,                          ! FAB block
;  302          0365  2          XAB_BLOCK : $XABFHC_DECL,                       ! XAB block
;  303          0366  2          FAB: REF BLOCK [, BYTE],                        ! Pointer to FAB
;  304          0367  2          NAM: REF BLOCK [, BYTE],                        ! Pointer to NAM
;  305          0368  2          UNIT,                                           ! LUN number
;  306          0369  2          RES_OR_EXP_NAME : VECTOR [NAM$C_MAXRSS, BYTE],  ! RSN or ESN
;  307          0370  2          RES_OR_EXP_LEN,                                 ! Length of resultant name
;  308          0371  2          INQ_FLAGS: BLOCK [4,BYTE] FIELD (INQ_FIELDS),   ! Internal flags
;  309          0372  2          VAR_LENGTHS : VECTOR [INQ$K_KEY_MAX + 1, BYTE], ! Length in bits of variables
;  310          0373  2          RESP_VEC : VECTOR [INQ$K_KEY_MAX + 1, LONG],    ! Response vector
;  311          0374  2          I,                                              ! Loop index
;  312          0375  2          RET_STATUS,                                     ! Returned error code from FOR$INQUIRE
;  313          0376  2          STATUS;                                         ! Returned condition code
;  314          0377  2
;  315          0378  2      BIND
;  316          0379  2          CCB_FAB = CCB: REF $FOR$FAB_CCB_STRUCT,
;  317          0380  2          CCB_NAM = CCB: REF $FOR$NAM_CCB_STRUCT;
;  318          0381  2
;  319          0382  2      ENABLE
;  320          0383  2          FOR$$ERR_OPECLO (L_UNWIND_ACTION, INQUIRE);     ! Set up error handler
;  321          0384  2
;  322          0385  2      !+
;  323          0386  2      ! Set up FAB, NAM and XAB blocks.
;  324          0387  2      !-
;  325          0388  2
;  326        P 0389  2      $FAB_INIT (FAB=FAB_BLOCK, NAM=NAM_BLOCK, XAB=XAB_BLOCK, DNM='.DAT',
;  327          0390  2          SHR=(GET,PUT,DEL,UPD,UPI));
;  328        P 0391  2      $NAM_INIT (NAM=NAM_BLOCK, ESA=RES_OR_EXP_NAME, ESS=NAM$C_MAXRSS,
;  329          0392  2          RSA=RES_OR_EXP_NAME, RSS=NAM$C_MAXRSS);
;  330          0393  2      $XABFHC_INIT (XAB=XAB_BLOCK);
;  331          0394  2
;  332          0395  2      !+
;  333          0396  2      ! Initialize internal flags and return status value.
;  334          0397  2      !-
;  335          0398  2      CH$FILL (0, (INQ$K_KEY_MAX +1) * %UPVAL, RESP_VEC);
;  336          0399  2      CH$FILL (0, INQ$K_KEY_MAX + 1, VAR_LENGTHS);
;  337          0400  2      INQ_FLAGS [NAME_OK] = 0;
;  338          0401  2      INQ_FLAGS [UNIT_OK] = 0;
;  339          0402  2      INQ_FLAGS [CCB_OK] = 0;
;  340          0403  2      INQ_FLAGS [EXISTS] = 0;
;  341          0404  2      INQ_FLAGS [FAB_OK] = 0;
;  342          0405  2      INQ_FLAGS [BY_FILE] = 0;
;  343          0406  2      RET_STATUS = 1;
;  344          0407  2      RES_OR_EXP_LEN = 0;          ! No name available yet
;  345          0408  2
;  346          0409  2      !+
;  347          0410  2      ! Set UNWIND cleanup to be a no-op since LUB/ISB/RAB
;  348          0411  2      ! has not been pushed yet.
;  349          0412  2      !-
;  350          0413  2
;  351          0414  2      L_UNWIND_ACTION = FOR$K_UNWINDNOP;
;  352          0415  2
;  353          0416  2      !+
;  354          0417  2      ! Copy keyword argument list into array INQUIRE
;  355          0418  2      ! in canonical order.  If FILE= is ASCIZ string, NAM_DSC is set
```

```
;  356     0419  2      ! up as the name descriptor.  May signal FOR$_INVARGFOR after
;  357     0420  2      ! setup.
;  358     0421  2      !-
;  359     0422  2
;  360     0423  2      FOR$$OPECLO_ARG (KEYWD, ACTUALCOUNT (), INQUIRE, INQ$K_KEY_MAX, NAM_DSC,
;  361     0424  2          DEF_DSC, 0, VAR_LENGTHS);
;  362     0425  2
;  363     0426  2      !+
;  364     0427  2      ! Mark IOSTAT as word or longword.  This is done here so that
;  365     0428  2      ! the general variable storage algorithm will work for IOSTAT
;  366     0429  2      ! without penalizing OPEN and CLOSE by doing the work in
;  367     0430  2      ! FOR$$OPECLO_ARG.
;  368     0431  2      !-
;  369     0432  2      IF .INQUIRE [INQ$K_IOSTAT] NEQ 0
;  370     0433  2      THEN
;  371     0434  2          IF .INQUIRE [INQ$K_IOSTAT_L]
;  372     0435  2          THEN
;  373     0436  2              VAR_LENGTHS [INQ$K_IOSTAT] = 32
;  374     0437  2          ELSE
;  375     0438  2              VAR_LENGTHS [INQ$K_IOSTAT] = 16;
;  376     0439  2
;  377     0440  2      !+
;  378     0441  2      ! Initially, point FAB and NAM at our local blocks.
;  379     0442  2      !-
;  380     0443  2
;  381     0444  2      FAB = FAB_BLOCK;
;  382     0445  2      NAM = NAM_BLOCK;
;  383     0446  2
;  384     0447  2      !+
;  385     0448  2      ! Now we split depending on whether this is an INQUIRE by FILE or
;  386     0449  2      ! an INQUIRE by UNIT.  If either FILE or DEFAULTFILE is present, it
;  387     0450  2      ! is an INQUIRE by FILE.  Otherwise it is an INQUIRE by UNIT.
;  388     0451  2      !-
;  389     0452  2
;  390     0453  2      IF .INQUIRE [INQ$K_FILE] NEQA 0 OR .INQUIRE [INQ$K_DEFAULTF] NEQA 0
;  391     0454  2      THEN
;  392     0455  3          BEGIN
;  393     0456  3
;  394     0457  3          !+
;  395     0458  3          ! This is INQUIRE by FILE.  Put the specified filename and/or
;  396     0459  3          ! default file name in the FAB and do a $OPEN to both see if the
;  397     0460  3          ! file exists and to get a resultant name string.  Then $CLOSE
;  398     0461  3          ! the file, because we no longer need it.
;  399     0462  3          !-
;  400     0463  3
;  401     0464  3          INQ_FLAGS [BY_FILE] = 1;
;  402     0465  3          INQ_FLAGS [NAME_OK] = 1;        ! Initially assume name is ok
;  403     0466  3
;  404     0467  3          IF .INQUIRE [INQ$K_DEFAULTF] NEQA 0
;  405     0468  3          THEN
;  406     0469  4              BEGIN
;  407     0470  4              LOCAL
;  408     0471  4                  DNAME: REF DSC$DESCRIPTOR;      ! Default filename descriptor
;  409     0472  4              DNAME = .INQUIRE [INQ$K_DEFAULTF];  ! Get file name address
;  410     0473  4              IF .DNAME [DSC$W_LENGTH] LEQU 255
;  411     0474  4              THEN
;  412     0475  5                  BEGIN
```

```
;  413     0476  5                FAB [FAB$B_DNS] = .DNAME [DSC$W_LENGTH];
;  414     0477  5                FAB [FAB$L_DNA] = .DNAME [DSC$A_POINTER];
;  415     0478  5                END
;  416     0479  4            ELSE
;  417     0480  4                INQ_FLAGS [NAME_OK] = 0;        ! Name is invalid
;  418     0481  3            END;
;  419     0482  3
;  420     0483  3        IF .INQUIRE [INQ$K_FILE] NEQA 0
;  421     0484  3        THEN
;  422     0485  4            BEGIN
;  423     0486  4            LOCAL
;  424     0487  4                FNAME: REF DSC$DESCRIPTOR;      ! Filename descriptor
;  425     0488  4            FNAME = .INQUIRE [INQ$K_FILE];      ! Get file name address
;  426     0489  4            IF .FNAME [DSC$W_LENGTH] LEQU 255
;  427     0490  4            THEN
;  428     0491  5                BEGIN
;  429     0492  5                FAB [FAB$B_FNS] = .FNAME [DSC$W_LENGTH];
;  430     0493  5                FAB [FAB$L_FNA] = .FNAME [DSC$A_POINTER];
;  431     0494  5                END
;  432     0495  4            ELSE
;  433     0496  4                INQ_FLAGS [NAME_OK] = 0;        ! Name is invalid
;  434     0497  3            END;
;  435     0498  3
;  436     0499  3        !+
;  437     0500  3        ! Do a $PARSE and $SEARCH to see if the name is ok.
;  438     0501  3        !-
;  439     0502  3        IF .INQ_FLAGS [NAME_OK]
;  440     0503  3        THEN
;  441     0504  4            IF $PARSE (FAB=FAB [0,0,0,0])
;  442     0505  3            THEN
;  443     0506  4                BEGIN
;  444     0507  6                IF (.NAM [NAM$L_FNB] AND (      ! Disallow wildcards
;  445     0508  6                    NAM$M_WILD_DIR OR
;  446     0509  6                    NAM$M_WILD_NAME OR
;  447     0510  6                    NAM$M_WILD_TYPE OR
;  448     0511  4                    NAM$M_WILD_VER)) NEQ 0
;  449     0512  4                THEN
;  450     0513  4                    INQ_FLAGS [NAME_OK] = 0
;  451     0514  4                ELSE
;  452     0515  5                    BEGIN
;  453     0516  5                    !+
;  454     0517  5                    ! Don't do $SEARCH for non-file-oriented devices.
;  455     0518  5                    !-
;  456     0519  5                    IF .BLOCK [FAB [FAB$L_DEV], DEV$V_FOD;4, BYTE]
;  457     0520  5                    THEN
;  458     0521  6                        $SEARCH (FAB=FAB [0,0,0,0])
;  459     0522  5                    ELSE
;  460     0523  5                        NAM [NAM$B_RSL] = .NAM [NAM$B_ESL]; ! Use ESN instead
;  461     0524  4                    END;
;  462     0525  4                END
;  463     0526  3            ELSE
;  464     0527  3                INQ_FLAGS [NAME_OK] = 0;
;  465     0528  3
;  466     0529  3
;  467     0530  3        IF .INQ_FLAGS [NAME_OK] ! No errors so far?
;  468     0531  3        THEN
;  469     0532  4            BEGIN
```

```
;  470     0533  4              IF .NAM [NAM$B_ESL] NEQ 0
;  471     0534  4              THEN
;  472     0535  4                  INQ_FLAGS [NAME_OK] = 1;        ! It's a vaild filename
;  473     0536  4              IF .NAM [NAM$B_RSL] NEQ 0
;  474     0537  4              THEN
;  475     0538  4                  INQ_FLAGS [EXISTS] = 1; ! File exists
;  476     0539  3              END;
;  477     0540  3
;  478     0541  3          FAB [FAB$V_NAM] = 1;
;  479     0542  3
;  480     0543  3          !+
;  481     0544  3          ! Now attempt to $OPEN the file.  We may fail for several
;  482     0545  3          ! reasons, one of which is that someone else, maybe us, has
;  483     0546  3          ! the file locked.  We try to recover from errors as gracefully
;  484     0547  3          ! as we can.
;  485     0548  3          !-
;  486     0549  3
;  487     0550  3          IF .INQ_FLAGS [EXISTS]
;  488     0551  3          THEN
;  489     0552  4              IF $OPEN (FAB=FAB [0,0,0,0])
;  490     0553  3              THEN
;  491     0554  4                  BEGIN
;  492     0555  4                  $CLOSE (FAB=FAB [0,0,0,0]);
;  493     0556  4                  INQ_FLAGS [FAB_OK] = 1;
;  494     0557  4                  END
;  495     0558  3              ELSE
;  496     0559  3                  INQ_FLAGS [EXISTS] = 0; ! If we can't open it, it doesn't exist.
;  497     0560  3
;  498     0561  3          !+
;  499     0562  3          ! Use Resultant name string or Expanded name string, in order
;  500     0563  3          ! of preference.
;  501     0564  3          !-
;  502     0565  3          RES_OR_EXP_LEN = .NAM [NAM$B_RSL];      ! Get length of result
;  503     0566  3          IF .RES_OR_EXP_LEN EQL 0
;  504     0567  3          THEN
;  505     0568  3              RES_OR_EXP_LEN = .NAM [NAM$B_ESL];
;  506     0569  3
;  507     0570  3          !+
;  508     0571  3          ! RMS expects us to keep doing $SEARCHs until no more files are
;  509     0572  3          ! found.  Well, we're not going to do that.  The problem is that
;  510     0573  3          ! RMS has allocated an IFAB (internal FAB) for the $SEARCH sequence
;  511     0574  3          ! and, if the file is on a remote node, has a FAL task waiting for the
;  512     0575  3          ! next search.  There is no clearly stated way of causing RMS to clean
;  513     0576  3          ! up.  The way we will do it is to do another $PARSE on the
;  514     0577  3          ! FAB after we have zeroed the FNM,ESA and RSA pointers.
;  515     0578  3          ! This is to prevent RMS from overwriting them.
;  516     0579  3          !-
;  517     0580  3
;  518     0581  3          IF .INQ_FLAGS [NAME_OK]
;  519     0582  3          THEN
;  520     0583  4              BEGIN
;  521     0584  4              FAB [FAB$L_FNA] = 0;
;  522     0585  4              FAB [FAB$B_FNS] = 0;
;  523     0586  4              FAB [FAB$L_DNA] = 0;
;  524     0587  4              NAM [NAM$L_ESA] = 0;
;  525     0588  4              NAM [NAM$L_RSA] = 0;
;  526     0589  4              $PARSE (FAB=FAB [0,0,0,0]);
```

```
;  527     0590  3          END;
;  528     0591  3
;  529     0592  3      IF .INQ_FLAGS [NAME_OK]
;  530     0593  3      THEN
;  531     0594  4          BEGIN
;  532     0595  4          !+
;  533     0596  4          ! If we successfully opened the file, all the necessary info
;  534     0597  4          ! is now in the FAB.  We have to scan the logical units to see
;  535     0598  4          ! if we have the file open, if we couldn't open the file.
;  536     0599  4          !-
;  537     0600  4
;  538     0601  4          LOCAL
;  539     0602  4              LUN_FLAG;                       ! Flag to OTS$$NEXT_LUN
;  540     0603  4
;  541     0604  4          !+
;  542     0605  4          ! Restore resultant or expanded name string
;  543     0606  4          !-
;  544     0607  4
;  545     0608  4          NAM [NAM$L_RSA] = RES_OR_EXP_NAME;
;  546     0609  4          NAM [NAM$B_RSL] = .RES_OR_EXP_LEN;
;  547     0610  4
;  548     0611  4          !+
;  549     0612  4          ! Begin scan of allocated logical units.
;  550     0613  4          !-
;  551     0614  4          LUN_FLAG = 0;                       ! Initialize LUN_FLAG
;  552     0615  4          IF .RET_STATUS THEN                 ! If no error so far
;  553     0616  4          DO                                  ! Until no more units
;  554     0617  5              BEGIN
;  555     0618  5              FOR$$NEXT_LUN (LUN_FLAG, UNIT); ! Get next used LUN
;  556     0619  5              IF .LUN_FLAG NEQ 0
;  557     0620  5              THEN
;  558     0621  6                  BEGIN
;  559     0622  6                  !+
;  560     0623  6                  ! We have a unit which has been allocated by FORTRAN.
;  561     0624  6                  ! We call FOR$$CB_FETCH to fetch the CCB.  If the
;  562     0625  6                  ! unit is opened and the names match then we use it,
;  563     0626  6                  ! else we try again.  This matching must be done while
;  564     0627  6                  ! ASTs are disabled so as to prevent someone from
;  565     0628  6                  ! playing in the LUB while we are deciding.
;  566     0629  6                  !-
;  567     0630  6                  LOCAL
;  568     0631  6                      AST_STATUS;     ! Returned from $SETAST
;  569     0632  6                  AST_STATUS = $SETAST (ENBFLG = 0);  ! Disable ASTs
;  570     0633  6                  FOR$$CB_FETCH (.UNIT);      ! Fetch the CCB for this unit
;  571     0634  6                  IF .CCB NEQ 0
;  572     0635  6                  THEN
;  573     0636  6                      IF .CCB [LUB$V_OPENED]
;  574     0637  6                      THEN
;  575     0638  6                          IF CH$EQL (.CCB [LUB$B_RSL], .CCB [LUB$A_RSN],
;  576     0639  6                              .RES_OR_EXP_LEN, RES_OR_EXP_NAME, %C' ')
;  577     0640  6                          THEN
;  578     0641  7                              BEGIN
;  579     0642  7                              !+
;  580     0643  7                              ! We have a match.  Call PUSH_CCB to
;  581     0644  7                              ! push the unit and return any errors
;  582     0645  7                              ! as its value.  ASTs are still disabled.
;  583     0646  7                              !-
```

```
;  584     0647  7                        INQ_FLAGS [UNIT_OK] = 1;
;  585     0648  7                        STATUS = PUSH_CCB (.UNIT);
;  586     0649  7                        IF .AST_STATUS EQL SS$_WASSET
;  587     0650  7                        THEN
;  588     0651  7                            $SETAST (ENBFLG = 1);       ! Reenable ASTs
;  589     0652  7                        IF .STATUS
;  590     0653  7                        THEN
;  591     0654  8                            BEGIN
;  592     0655  8                            L_UNWIND_ACTION = FOR$K_UNWINDPOP;
;  593     0656  8                            INQ_FLAGS [CCB_OK] = 1;
;  594     0657  8                            INQ_FLAGS [EXISTS] = 1;
;  595     0658  8                            EXITLOOP;
;  596     0659  8                            END
;  597     0660  7                        ELSE
;  598     0661  8                            BEGIN
;  599     0662  8                            !+
;  600     0663  8                            ! The push failed.  Just
;  601     0664  8                            ! exit the loop.
;  602     0665  8                            !-
;  603     0666  8                            EXITLOOP;
;  604     0667  7                            END;
;  605     0668  6                        END;
;  606     0669  6                    !+
;  607     0670  6                    ! No match.  Continue scanning units.
;  608     0671  6                    !-
;  609     0672  6                    IF .AST_STATUS EQL SS$_WASSET
;  610     0673  6                    THEN
;  611     0674  6                        $SETAST (ENBFLG = 1);   ! Reenable ASTs
;  612     0675  5                    END;
;  613     0676  5                END
;  614     0677  4            UNTIL .LUN_FLAG EQL 0;      ! Until no more LUNs
;  615     0678  4
;  616     0679  4                END
;  617     0680  3            ELSE
;  618     0681  4                BEGIN
;  619     0682  4                !+
;  620     0683  4                ! Name is invalid.  Point "resultant name" at filename.
;  621     0684  4                !-
;  622     0685  4
;  623     0686  4                NAM [NAM$L_RSA] = .FAB [FAB$L_FNA];
;  624     0687  4                NAM [NAM$B_RSL] = .FAB [FAB$B_FNS];
;  625     0688  3                END;
;  626     0689  3            END
;  627     0690  3
;  628     0691  2        ELSE
;  629     0692  2
;  630     0693  3            BEGIN
;  631     0694  3            !+
;  632     0695  3            ! This is INQUIRE by UNIT.
;  633     0696  3            !-
;  634     0697  3
;  635     0698  3            UNIT = .INQUIRE [INQ$K_UNIT];   ! Get unit number
;  636     0699  3            IF .UNIT GEQ LUB$K_LUN_MIN AND .UNIT LEQ LUB$K_LUN_MAX  ! Unit in range 0-99?
;  637     0700  3            THEN
;  638     0701  4                BEGIN
;  639     0702  4                INQ_FLAGS [UNIT_OK] = 1;
;  640     0703  4                !+
```

```
;  641     0704  4          ! We know that the unit is a valid number, but we don't
;  642     0705  4          ! know if it has been opened by FORTRAN.  Call PUSH_CCB
;  643     0706  4          ! to attempt the push.  It may fail because the unit was
;  644     0707  4          ! not allocated by FORTRAN or because of recursive I/O,
;  645     0708  4          ! or other reasons.
;  646     0709  4          !-
;  647     0710  4          STATUS = PUSH_CCB (.UNIT);
;  648     0711  4          IF .STATUS
;  649     0712  4          THEN
;  650     0713  5              BEGIN
;  651     0714  5              !+
;  652     0715  5              ! Success.  Use this CCB.
;  653     0716  5              !-
;  654     0717  5              L_UNWIND_ACTION = FOR$K_UNWINDPOP;
;  655     0718  5              IF .CCB [LUB$V_OPENED]  ! Unit open?
;  656     0719  5              THEN
;  657     0720  6                  BEGIN
;  658     0721  6                  INQ_FLAGS [CCB_OK] = 1;
;  659     0722  6                  INQ_FLAGS [EXISTS] = 1;
;  660     0723  6                  INQ_FLAGS [NAME_OK] = 1;
;  661     0724  6                  END
;  662     0725  5              ELSE
;  663     0726  6                  BEGIN
;  664     0727  6                  !+
;  665     0728  6                  ! The unit is not open.  Return it and try again.
;  666     0729  6                  !-
;  667     0730  6                  FOR$$CB_POP ();      !  Return the LUB
;  668     0731  6                  L_UNWIND_ACTION = FOR$K_UNWINDNOP;
;  669     0732  5                  END;
;  670     0733  5              END
;  671     0734  4          ELSE
;  672     0735  4              !+
;  673     0736  4              ! The push failed.
;  674     0737  4              !-
;  675     0738  4              RET_STATUS = .STATUS;
;  676     0739  3          END;
;  677     0740  2      END;
;  678     0741  2
```

```
;  680          0742  2      !+
;  681          0743  2      ! If the CCB is valid, point the FAB and NAM pointers at the blocks
;  682          0744  2      ! in the CCB.
;  683          0745  2      !-
;  684          0746  2
;  685          0747  2      IF .INQ_FLAGS [CCB_OK]
;  686          0748  2      THEN
;  687          0749  3          BEGIN
;  688          0750  3          FAB = CCB_FAB [0,0,0,0];
;  689          0751  3          NAM = CCB_NAM [0,0,0,0];
;  690          0752  2          END;
;  691          0753  2
;  692          0754  2      !+
;  693          0755  2      ! Now fill in return values.
;  694          0756  2      !-
;  695          0757  2
;  696          0758  2      !+
;  697          0759  2      ! EXIST - Logical variable.
;  698          0760  2      !    By file - TRUE if file exists, otherwise FALSE.
;  699          0761  2      !    By unit - TRUE if unit exists (0-99), otherwise FALSE.
;  700          0762  2      !-
;  701          0763  2      IF .INQUIRE [INQ$K_EXIST] NEQ 0
;  702          0764  2      THEN
;  703          0765  3          IF .INQ_FLAGS [EXISTS] OR ((NOT .INQ_FLAGS [BY_FILE]) AND
;  704          0766  3              .INQ_FLAGS [UNIT_OK])
;  705          0767  2          THEN
;  706          0768  2              RESP_VEC [INQ$K_EXIST] = -1
;  707          0769  2          ELSE
;  708          0770  2              RESP_VEC [INQ$K_EXIST] = 0;
;  709          0771  2
;  710          0772  2      !+
;  711          0773  2      ! OPENED - logical variable
;  712          0774  2      !    TRUE if unit is opened, otherwise FALSE.
;  713          0775  2      !-
;  714          0776  2      IF .INQUIRE [INQ$K_OPENED] NEQ 0
;  715          0777  2      THEN
;  716          0778  2          IF (.INQ_FLAGS [CCB_OK]) OR     ! True if unit connected to a file
;  717          0779  3             (.INQ_FLAGS [BY_FILE] AND .INQ_FLAGS [UNIT_OK])
;  718          0780  2          THEN
;  719          0781  2              RESP_VEC [INQ$K_OPENED] = -1
;  720          0782  2          ELSE
;  721          0783  2              RESP_VEC [INQ$K_OPENED] = 0;
;  722          0784  2
;  723          0785  2      !+
;  724          0786  2      ! NUMBER - integer variable
;  725          0787  2      !    By file - unit number that file is connected to.
;  726          0788  2      !    By unit - returns unit number if connected.
;  727          0789  2      !-
;  728          0790  2
;  729          0791  2      IF .INQUIRE [INQ$K_NUMBER] NEQ 0
;  730          0792  2      THEN
;  731          0793  2          IF .INQ_FLAGS [UNIT_OK]
;  732          0794  2          THEN
;  733          0795  2              RESP_VEC [INQ$K_NUMBER] = .UNIT
;  734          0796  2          ELSE
;  735          0797  2              RESP_VEC [INQ$K_NUMBER] = 0;
;  736          0798  2
```

```
;  737     0799  2      !+
;  738     0800  2      ! NAMED - logical variable
;  739     0801  2      !   If file has a name, then this is TRUE.  If file is opened SCRATCH,
;  740     0802  2      !   then it is considered not to be named.
;  741     0803  2      !-
;  742     0804  2      IF .INQUIRE [INQ$K_NAMED] NEQ 0
;  743     0805  2      THEN
;  744     0806  3          BEGIN
;  745     0807  3          IF .INQ_FLAGS [NAME_OK]
;  746     0808  3          THEN
;  747     0809  3              RESP_VEC [INQ$K_NAMED] = -1
;  748     0810  3          ELSE
;  749     0811  3              RESP_VEC [INQ$K_NAMED] = 0;
;  750     0812  3          IF .INQ_FLAGS [CCB_OK] THEN IF .CCB [LUB$V_SCRATCH]
;  751     0813  3          THEN
;  752     0814  3              RESP_VEC [INQ$K_NAMED] = 0;
;  753     0815  2          END;
;  754     0816  2
;  755     0817  2      !+
;  756     0818  2      ! NAME - character variable
;  757     0819  2      !   If NAMED would be true, then the fully qualified file name that results.
;  758     0820  2      !   This file might not exist, but the filename is valid.
;  759     0821  2      !-
;  760     0822  2      IF .INQUIRE [INQ$K_NAME] NEQ 0
;  761     0823  2      THEN
;  762     0824  3          BEGIN
;  763     0825  3          LOCAL
;  764     0826  3              NAME_DSC : REF DSC$DESCRIPTOR;
;  765     0827  3          NAME_DSC = .INQUIRE [INQ$K_NAME];
;  766     0828  3          CH$COPY (.NAM [NAM$B_RSL], .NAM [NAM$L_RSA], %C' ',
;  767     0829  3              .NAME_DSC [DSC$W_LENGTH], .NAME_DSC [DSC$A_POINTER]);
;  768     0830  2          END;
;  769     0831  2
;  770     0832  2      !+
;  771     0833  2      ! ACCESS - character variable
;  772     0834  2      !   Access type specified by OPEN or DEFINE FILE.  Can be 'SEQUENTIAL',
;  773     0835  2      !   'DIRECT', 'KEYED' or 'UNKNOWN' if not connected.
;  774     0836  2      !-
;  775     0837  2      IF .INQUIRE [INQ$K_ACCESS] NEQ 0
;  776     0838  2      THEN
;  777     0839  3          BEGIN
;  778     0840  3          RESP_VEC [INQ$K_ACCESS] = UNKNOWN;
;  779     0841  3          IF .INQ_FLAGS [CCB_OK]
;  780     0842  3          THEN
;  781     0843  3              IF .CCB [LUB$V_DIRECT]
;  782     0844  3              THEN
;  783     0845  3                  RESP_VEC [INQ$K_ACCESS] = DIRECT
;  784     0846  3              ELSE IF .CCB [LUB$V_KEYED]
;  785     0847  3              THEN
;  786     0848  3                  RESP_VEC [INQ$K_ACCESS] = KEYED
;  787     0849  3              ELSE
;  788     0850  3                  RESP_VEC [INQ$K_ACCESS] = SEQUENTIAL
;  789     0851  3          ELSE
;  790     0852  3              RESP_VEC [INQ$K_ACCESS] = UNKNOWN;
;  791     0853  2          END;
;  792     0854  2
;  793     0855  2      !+
```

```
;  794          0856  2      ! SEQUENTIAL - character variable
;  795          0857  2      !    If ACCESS='SEQUENTIAL' is allowed for this file, then this is 'YES'.
;  796          0858  2      !    An answer of 'NO' is impossible for VAX, since ALL files can be accessed
;  797          0859  2      !    sequentially.  However, if we can't open the file, we return 'UNKNOWN'.
;  798          0860  2      !-
;  799          0861  2      IF .INQUIRE [INQ$K_SEQUENTIA] NEQ 0
;  800          0862  2      THEN
;  801          0863  2          IF .INQ_FLAGS [EXISTS]
;  802          0864  2          THEN
;  803          0865  2              RESP_VEC [INQ$K_SEQUENTIA] = YES
;  804          0866  2          ELSE
;  805          0867  2              RESP_VEC [INQ$K_SEQUENTIA] = UNKNOWN;
;  806          0868  2
;  807          0869  2      !+
;  808          0870  2      ! DIRECT - character variable
;  809          0871  2      !    If ACCESS='DIRECT' is allowed for this file, then we answer 'YES'.
;  810          0872  2      !    If not allowed, answer 'NO'.  If we can't open the file, answer 'UNKNOWN'.
;  811          0873  2      !-
;  812          0874  2      IF .INQUIRE [INQ$K_DIRECT] NEQ 0
;  813          0875  2      THEN
;  814          0876  3          BEGIN
;  815          0877  3          RESP_VEC [INQ$K_DIRECT] = UNKNOWN;
;  816          0878  3          IF .INQ_FLAGS [EXISTS]
;  817          0879  3          THEN
;  818          0880  4              BEGIN
;  819          0881  4              SELECTONE .FAB [FAB$B_ORG] OF
;  820          0882  4                  SET
;  821          0883  4
;  822          0884  4                  [FAB$C_SEQ] :
;  823          0885  4                      IF .FAB [FAB$B_RFM] EQL FAB$C_FIX
;  824          0886  4                      THEN
;  825          0887  4                          RESP_VEC [INQ$K_DIRECT] = YES
;  826          0888  4                      ELSE
;  827          0889  4                          RESP_VEC [INQ$K_DIRECT] = NO;
;  828          0890  4                  [FAB$C_REL] :
;  829          0891  4                      RESP_VEC [INQ$K_DIRECT] = YES;
;  830          0892  4                  [FAB$C_IDX] :
;  831          0893  4                      RESP_VEC [INQ$K_DIRECT] = NO;
;  832          0894  4
;  833          0895  4                  TES;
;  834          0896  3              END;
;  835          0897  2          END;
;  836          0898  2
;  837          0899  2      !+
;  838          0900  2      ! KEYED - character variable
;  839          0901  2      !    Is ACCESS='KEYED' allowed for this file?  'YES' if it's INDEXED
;  840          0902  2      !    organization, 'NO' if not and 'UNKNOWN' if we can't tell.
;  841          0903  2      !-
;  842          0904  2      IF .INQUIRE [INQ$K_KEYED] NEQ 0
;  843          0905  2      THEN
;  844          0906  3          BEGIN
;  845          0907  3          RESP_VEC [INQ$K_KEYED] = UNKNOWN;
;  846          0908  3          IF .INQ_FLAGS [EXISTS]
;  847          0909  3          THEN
;  848          0910  4              BEGIN
;  849          0911  4              SELECTONE .FAB [FAB$B_ORG] OF
;  850          0912  4                  SET
```

```
;  851     0913  4
;  852     0914  4                 [FAB$C_IDX] :
;  853     0915  4                     RESP_VEC [INQ$K_KEYED] = YES;
;  854     0916  4                 [FAB$C_SEQ,FAB$C_RE[] :
;  855     0917  4                     RESP_VEC [INQ$K_KEYED] = NO;
;  856     0918  4
;  857     0919  4                 TES;
;  858     0920  3             END;
;  859     0921  2         END;
;  860     0922  2
;  861     0923  2         !+
;  862     0924  2         ! FORM - character variable
;  863     0925  2         !    If the file is connected for FORMATTED or UNFORMATTED I/O, then
;  864     0926  2         !    return the string 'FORMATTED' or 'UNFORMATTED' as is appropriate.
;  865     0927  2         !    If we can't tell, return 'UNKNOWN'.
;  866     0928  2         !-
;  867     0929  2
;  868     0930  2         IF .INQUIRE [INQ$K_FORM] NEQ 0
;  869     0931  2         THEN
;  870     0932  3             BEGIN
;  871     0933  3             RESP_VEC [INQ$K_FORM] = UNKNOWN;
;  872     0934  3             IF .INQ_FLAGS [CCB_OK]
;  873     0935  3             THEN
;  874     0936  3                 IF .CCB [LUB$V_FORMATTED]
;  875     0937  3                 THEN
;  876     0938  3                     RESP_VEC [INQ$K_FORM] = FORMATTED
;  877     0939  3                 ELSE IF .CCB [LUB$V_UNFORMAT]
;  878     0940  3                 THEN
;  879     0941  3                     RESP_VEC [INQ$K_FORM] = UNFORMATTED;
;  880     0942  2             END;
;  881     0943  2
;  882     0944  2         !+
;  883     0945  2         ! FORMATTED - character variable
;  884     0946  2         !    If FORM='FORMATTED' allowed?  If so, answer 'YES'.  There is
;  885     0947  2         !    no way to answer 'NO'.
;  886     0948  2         !    'UNKNOWN' if we can't tell.
;  887     0949  2         !-
;  888     0950  2
;  889     0951  2         IF .INQUIRE [INQ$K_FORMATTED] NEQ 0
;  890     0952  2         THEN
;  891     0953  2             IF .INQ_FLAGS [EXISTS]
;  892     0954  2             THEN
;  893     0955  2                 RESP_VEC [INQ$K_FORMATTED] = YES
;  894     0956  2             ELSE
;  895     0957  2                 RESP_VEC [INQ$K_FORMATTED] = UNKNOWN;
;  896     0958  2
;  897     0959  2
;  898     0960  2         !+
;  899     0961  2         ! UNFORMATTED - character variable
;  900     0962  2         !    Is FORM='UNFORMATTED' allowed?  If so, answer 'YES'.
;  901     0963  2         !    'NO' is impossible.  'UNKNOWN' if we can't tell.
;  902     0964  2         !-
;  903     0965  2
;  904     0966  2         IF .INQUIRE [INQ$K_UNFORMATT] NEQ 0
;  905     0967  2         THEN
;  906     0968  2             IF .INQ_FLAGS [EXISTS]
;  907     0969  2             THEN
```

```
;  908     0970  2              RESP_VEC [INQ$K_UNFORMATT] = YES
;  909     0971  2          ELSE
;  910     0972  2              RESP_VEC [INQ$K_UNFORMATT] = UNKNOWN;
;  911     0973  2
;  912     0974  2      !+
;  913     0975  2      ! ORGANIZATION - character string
;  914     0976  2      !   Return the file organization as 'SEQUENTIAL', 'RELATIVE' or
;  915     0977  2      !   'INDEXED'.  Return 'UNKNOWN' if we can't open the file.
;  916     0978  2      !-
;  917     0979  2
;  918     0980  2      IF .INQUIRE [INQ$K_ORGANIZAT] NEQ 0
;  919     0981  2      THEN
;  920     0982  2          BEGIN
;  921     0983  2          RESP_VEC [INQ$K_ORGANIZAT] = UNKNOWN;
;  922     0984  2          IF .INQ_FLAGS [EXISTS]
;  923     0985  2          THEN
;  924     0986  2              SELECTONE .FAB [FAB$B_ORG] OF
;  925     0987  2                  SET
;  926     0988  2
;  927     0989  2                  [FAB$C_SEQ] :
;  928     0990  2                      RESP_VEC [INQ$K_ORGANIZAT] = SEQUENTIAL;
;  929     0991  2                  [FAB$C_REL] :
;  930     0992  2                      RESP_VEC [INQ$K_ORGANIZAT] = RELATIVE;
;  931     0993  2                  [FAB$C_IDX] :
;  932     0994  2                      RESP_VEC [INQ$K_ORGANIZAT] = INDEXED;
;  933     0995  2
;  934     0996  2                  TES;
;  935     0997  2          END;
;  936     0998  2
;  937     0999  2      !+
;  938     1000  2      ! RECL - integer variable
;  939     1001  2      !   Return the record length of the file.  If the file is opened,
;  940     1002  2      !   the current length is taken.  Else if the file exists the
;  941     1003  2      !   size used is the MAX of FAB$W_MRS and XAB$W_LRL.
;  942     1004  2      !   The record length is in bytes unless the file is connected for
;  943     1005  2      !   UNFORMATTED, in which case the length is in longwords.  If
;  944     1006  2      !   the file is connected SEGMENTED, then 2 bytes are subtracted
;  945     1007  2      !   from the length.  This is the inverse of the calculations
;  946     1008  2      !   done by OPEN.  If the record length can not be determined,
;  947     1009  2      !   0 is returned.
;  948     1010  2      !-
;  949     1011  2      IF .INQUIRE [INQ$K_RECL] NEQ 0
;  950     1012  2      THEN
;  951     1013  3          BEGIN
;  952     1014  3          RESP_VEC [INQ$K_RECL] = 0;      ! If recordsize is undefined
;  953     1015  3          IF .INQ_FLAGS [CCB_OK]
;  954     1016  3          THEN
;  955     1017  4              BEGIN
;  956     1018  4              IF .CCB [LUB$W_RBUF_SIZE] NEQ 0
;  957     1019  4              THEN
;  958     1020  5                  BEGIN
;  959     1021  5                  RESP_VEC [INQ$K_RECL] = .CCB [LUB$W_RBUF_SIZE];
;  960     1022  5                  IF .CCB [LUB$V_SEGMENTED]
;  961     1023  5                  THEN
;  962     1024  5                      RESP_VEC [INQ$K_RECL] = .RESP_VEC [INQ$K_RECL] - 2;
;  963     1025  5                  IF .CCB [LUB$V_UNFORMAT]
;  964     1026  5                  THEN
```

```
;  965          1027  5                    RESP_VEC [INQ$K_RECL] = .RESP_VEC [INQ$K_RECL] / %UPVAL;
;  966          1028  4                END;
;  967          1029  4            END
;  968          1030  3        ELSE IF .INQ_FLAGS [FAB_OK]
;  969          1031  3        THEN
;  970          1032  3            RESP_VEC [INQ$K_RECL] = MAXU (.FAB [FAB$W_MRS],
;  971          1033  3                .XAB_BLOCK [XAB$W_LRL]);
;  972          1034  2        END;
;  973          1035  2
;  974          1036  2        !+
;  975          1037  2        ! NEXTREC - integer variable
;  976          1038  2        !    If the file is connected for direct access, return the next logical
;  977          1039  2        !    record number.  If it is not connected for direct access, return 0.
;  978          1040  2        !-
;  979          1041  2        IF .INQUIRE [INQ$K_NEXTREC] NEQ 0
;  980          1042  2        THEN
;  981          1043  3            BEGIN
;  982          1044  3            RESP_VEC [INQ$K_NEXTREC] = 0;
;  983          1045  3            IF .INQ_FLAGS [CCB_OK] THEN IF .CCB [LUB$V_DIRECT]
;  984          1046  3            THEN
;  985          1047  3                RESP_VEC [INQ$K_NEXTREC] = .CCB [LUB$L_LOG_RECNO];
;  986          1048  2            END;
;  987          1049  2
;  988          1050  2        !+
;  989          1051  2        ! BLANK - character variable
;  990          1052  2        !    If the file is connected for FORMATTED, return the BLANK=
;  991          1053  2        !    value in effect, either 'ZERO' or 'NULL'.  If we can't tell,
;  992          1054  2        !    return 'UNKNOWN'.
;  993          1055  2        !-
;  994          1056  2        IF .INQUIRE [INQ$K_BLANK] NEQ 0
;  995          1057  2        THEN
;  996          1058  3            BEGIN
;  997          1059  3            RESP_VEC [INQ$K_BLANK] = UNKNOWN;
;  998          1060  3            IF .INQ_FLAGS [CCB_OK] THEN IF .CCB [LUB$V_FORMATTED]
;  999          1061  3            THEN
; 1000          1062  4                BEGIN
; 1001          1063  4                IF .CCB [LUB$V_NULLBLNK]
; 1002          1064  4                THEN
; 1003          1065  4                    RESP_VEC [INQ$K_BLANK] = NULL
; 1004          1066  4                ELSE
; 1005          1067  4                    RESP_VEC [INQ$K_BLANK] = ZERO;
; 1006          1068  3                END;
; 1007          1069  2            END;
; 1008          1070  2
; 1009          1071  2        !+
; 1010          1072  2        ! RECORDTYPE - character variable
; 1011          1073  2        !    Return the file's recordtype.  'FIXED' or 'VARIABLE'.  Return
; 1012          1074  2        !    'SEGMENTED' if the file is currently connected for SEGMENTED.
; 1013          1075  2        !    'UNKNOWN' if we can't tell.
; 1014          1076  2        !-
; 1015          1077  2        IF .INQUIRE [INQ$K_RECORDTYP] NEQ 0
; 1016          1078  2        THEN
; 1017          1079  3            BEGIN
; 1018          1080  3            RESP_VEC [INQ$K_RECORDTYP] = UNKNOWN;
; 1019          1081  3            IF .INQ_FLAGS [EXISTS]
; 1020          1082  3            THEN
; 1021          1083  3                SELECTONEU .FAB [FAB$B_RFM] OF
```

```
;  1022     1084  3                    SET
;  1023     1085  3
;  1024     1086  3                    [FAB$C_FIX] :
;  1025     1087  3                        RESP_VEC [INQ$K_RECORDTYP] = FIXED;
;  1026     1088  3                    [FAB$C_VAR, FAB$C_VFC] :
;  1027     1089  3                        RESP_VEC [INQ$K_RECORDTYP] = VARIABLE;
;  1028     1090  3                    [FAB$C_STM] :
;  1029     1091  3                        RESP_VEC [INQ$K_RECORDTYP] = STREAM;
;  1030     1092  3                    [FAB$C_STMCR] :
;  1031     1093  3                        RESP_VEC [INQ$K_RECORDTYP] = STREAM_CR;
;  1032     1094  3                    [FAB$C_STMLF] :
;  1033     1095  3                        RESP_VEC [INQ$K_RECORDTYP] = STREAM_LF;
;  1034     1096  3                    TES;
;  1035     1097  3                IF .INQ_FLAGS [CCB_OK]
;  1036     1098  3                THEN
;  1037     1099  3                    IF .CCB [LUB$V_SEGMENTED]
;  1038     1100  3                    THEN
;  1039     1101  3                        RESP_VEC [INQ$K_RECORDTYP] = SEGMENTED;
;  1040     1102  2                END;
;  1041     1103  2
;  1042     1104  2            !+
;  1043     1105  2            ! CARRIAGECONTROL
;  1044     1106  2            !-
;  1045     1107  2            IF .INQUIRE [INQ$K_CARRIAGE] NEQ 0
;  1046     1108  2            THEN
;  1047     1109  3                BEGIN
;  1048     1110  3                RESP_VEC [INQ$K_CARRIAGE] = UNKNOWN;
;  1049     1111  3                IF .INQ_FLAGS [EXISTS]
;  1050     1112  3                THEN
;  1051     1113  4                    BEGIN
;  1052     1114  4                    IF .FAB [FAB$V_FTN]
;  1053     1115  4                    THEN
;  1054     1116  4                        RESP_VEC [INQ$K_CARRIAGE] = FORTRAN
;  1055     1117  4                    ELSE IF .FAB [FAB$V_CR]
;  1056     1118  4                    THEN
;  1057     1119  4                        RESP_VEC [INQ$K_CARRIAGE] = LIST
;  1058     1120  4                    ELSE IF NOT .FAB [FAB$V_PRN]
;  1059     1121  4                    THEN
;  1060     1122  4                        RESP_VEC [INQ$K_CARRIAGE] = NONE;
;  1061     1123  4                    END
;  1062     1124  2                END;
```

```
;  1064     1125  2      !+
;  1065     1126  2      ! Store all responses.
;  1066     1127  2      !-
;  1067     1128  2
;  1068     1129  2      !+
;  1069     1130  2      ! Loop through keyword table and store value.  Value may be
;  1070     1131  2      ! numeric, character or ignorable.
;  1071     1132  2      !-
;  1072     1133  2
;  1073     1134  2      INCR I FROM INQ$K_IOSTAT TO INQ$K_KEY_MAX DO
;  1074     1135  2          IF .INQUIRE [.I] NEQ 0
;  1075     1136  2          THEN
;  1076     1137  2              CASE .RESP_TYPES [.I] FROM 0 TO 2 OF
;  1077     1138  2                  SET
;  1078     1139  2
;  1079     1140  2                  [0] :
;  1080     1141  2                      ;   ! Do nothing.  Value is already stored.
;  1081     1142  2
;  1082     1143  2                  [1] :   ! Numeric value.
;  1083     1144  3                      BEGIN
;  1084     1145  3                      LOCAL
;  1085     1146  3                          DEST : REF BLOCK [,BYTE];
;  1086     1147  3                      DEST = .INQUIRE [.I];          ! Address of destination
;  1087     1148  3                      DEST [0, 0, .VAR_LENGTHS [.I], 1] = .RESP_VEC [.I];
;  1088     1149  3                      END;
;  1089     1150  2
;  1090     1151  2                  [2] :   ! Character value.
;  1091     1152  3                      BEGIN
;  1092     1153  3                      LOCAL
;  1093     1154  3                          DSC : REF DSC$DESCRIPTOR,
;  1094     1155  3                          WHICH;              !  Key of response value
;  1095     1156  3                      DSC = .INQUIRE [.I];           ! Result descriptor
;  1096     1157  3                      WHICH = .RESP_VEC [.I];
;  1097     1158  3                      CH$COPY (.RESP_LENS [.WHICH], .RESP_VALS [.WHICH] + RESP_VALS,
;  1098     1159  3                          %C' ', .DSC [DSC$W_LENGTH], .DSC [DSC$A_POINTER]);
;  1099     1160  2                      END;
;  1100     1161  2
;  1101     1162  2              TES;
```

```
2      !+
2      ! If we got an error, now's the time to signal it.
2      !-
2      IF NOT .RET_STATUS
2      THEN
2          FOR$$SIG_NO_LUB (.RET_STATUS, .UNIT);
2
2      !+
2      ! Pop the CCB if we have one.
2      !-
2      IF .INQ_FLAGS [CCB_OK]
2      THEN
2          FOR$$CB_POP ();
2
2
2      !+
2      ! Return success - we only get here if there has been no error
2      !-
2
2      RETURN 1;
1      END;
```

```
                                                                  .TITLE  FOR$INQUIRE FORTRAN INQUIRE
                                                                  .IDENT  \1-017\

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

                                    20  00000 P.AAA:  .ASCII  \ \                                     ;
                              53 45 59  00001 P.AAB:  .ASCII  \YES\                                   ;
                                 4F 4E  00004 P.AAC:  .ASCII  \NO\                                    ;
                  4E 57 4F 4E 4B 4E 55  00006 P.AAD:  .ASCII  \UNKNOWN\                               ;
                     54 43 45 52 49 44  0000D P.AAE:  .ASCII  \DIRECT\                                ;
                        44 45 59 45 4B  00013 P.AAF:  .ASCII  \KEYED\                                 ;
         4C 41 49 54 4E 45 55 51 45 53  00018 P.AAG:  .ASCII  \SEQUENTIAL\                            ;
            44 45 54 54 41 4D 52 4F 46  00022 P.AAH:  .ASCII  \FORMATTED\                             ;
      44 45 54 54 41 4D 52 4F 46 4E 55  0002B P.AAI:  .ASCII  \UNFORMATTED\                           ;
               45 56 49 54 41 4C 45 52  00036 P.AAJ:  .ASCII  \RELATIVE\                              ;
                  44 45 58 45 44 4E 49  0003E P.AAK:  .ASCII  \INDEXED\                               ;
                           4C 4C 55 4E  00045 P.AAL:  .ASCII  \NULL\                                  ;
                           4F 52 45 5A  00049 P.AAM:  .ASCII  \ZERO\                                  ;
                        44 45 58 49 46  0004D P.AAN:  .ASCII  \FIXED\                                 ;
               45 4C 42 41 49 52 41 56  00052 P.AAO:  .ASCII  \VARIABLE\                              ;
            44 45 54 4E 45 4D 47 45 53  0005A P.AAP:  .ASCII  \SEGMENTED\                             ;
                  4E 41 52 54 52 4F 46  00063 P.AAQ:  .ASCII  \FORTRAN\                               ;
                           54 53 49 4C  0006A P.AAR:  .ASCII  \LIST\                                  ;
                           45 4E 4F 4E  0006E P.AAS:  .ASCII  \NONE\                                  ;
                     4D 41 45 52 54 53  00072 P.AAT:  .ASCII  \STREAM\                                ;
            52 43 5F 4D 41 45 52 54 53  00078 P.AAU:  .ASCII  \STREAM_CR\                             ;
            46 4C 5F 4D 41 45 52 54 53  00081 P.AAV:  .ASCII  \STREAM_LF\                             ;
                                        0008A         .BLKB   2
00000000* 00000000* 00000000* 00000000* 00000000* 00000000* 0008C RESP_VALS:
                                                              .LONG   <P.AAA-RESP_VALS>, <P.AAB-RESP_VALS>, -   ;
00000000* 00000000* 00000000* 00000000* 00000000* 00000000* 000A4         <P.AAC-RESP_VALS>, <P.AAD-RESP_VALS>, -   ;
00000000* 00000000* 00000000* 00000000* 00000000* 00000000* 000BC         <P.AAE-RESP_VALS>, <P.AAF-RESP_VALS>, -   ;
                    00000000* 00000000* 00000000* 00000000* 000D4         <P.AAG-RESP_VALS>, <P.AAH-RESP_VALS>, -   ;
                                                                          <P.AAI-RESP_VALS>, <P.AAJ-RESP_VALS>, -   ;
```

```
                                                                  <P.AAK-RESP_VALS>, <P.AAL-RESP_VALS>, -          ;
                                                                  <P.AAM-RESP_VALS>, <P.AAN-RESP_VALS>, -          ;
                                                                  <P.AAO-RESP_VALS>, <P.AAP-RESP_VALS>, -          ;
                                                                  <P.AAQ-RESP_VALS>, <P.AAR-RESP_VALS>, -          ;
                                                                  <P.AAS-RESP_VALS>, <P.AAT-RESP_VALS>, -          ;
                                                                  <P.AAU-RESP_VALS>, <P.AAV-RESP_VALS>             ;
08 05 04 04 07 08 0B 09 0A 05 06 07 02 03 01 000E4 RESP_LENS:
                                                        .BYTE     1, 3, 2, 7, 6, 5, 10, 9, 11, 8, 7, 4, 4, -       ;
                        09 09 06 04 04 07 09 000F3                5, 8, 9, 7, 4, 4, 6, 9, 9                        ;
                                             000FA      .BLKB     2
                                         00# 000FC RESP_TYPES:
                                                        .BYTE     0[22]                                            ;
                                          01 00112      .BYTE     1                                                ;
                                         00# 00113      .BYTE     0[7]                                             ;
02 02 01 01 02 02 02 02 02 02 00 01 01 01 01 0011A      .BYTE     1, 1, 1, 1, 0, 2, 2, 2, 2, 2, 2, 1, 1, 2, -      ;
                                    02 02 02 00129                2, 2, 2, 2                                       ;
                                 54 41 44 2E 0012C P.AAW:  .ASCII \.DAT\                                           ;

                                                        .EXTRN    FOR$$ERR_OPECLO
                                                        .EXTRN    FOR$$ERRSNS_SAV
                                                        .EXTRN    FOR$$OPECLO_ARG
                                                        .EXTRN    FOR$$CB_PUSH, FOR$$CB_POP
                                                        .EXTRN    FOR$$CB_FETCH, FOR$$SIG_NO_LUB
                                                        .EXTRN    LIB$SIG_TO_RET, LIB$STOP
                                                        .EXTRN    FOR$$NEXT_LUN, SYS$PARSE
                                                        .EXTRN    SYS$SEARCH, SYS$OPEN
                                                        .EXTRN    SYS$CLOSE, SYS$SETAST

                                        0FFC 00000      .ENTRY    FOR$INQUIRE, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0305
                                                                  R10,R11
                       5E    FC58   CE   9E 00002       MOVAB     -936(SP), SP                                     ;
                             FF2C   CD   7C 00007       CLRQ      INQUIRE                                          ; 0347
                             FF34   CD   7C 0000B       CLRQ      INQUIRE                                          ;
                             FF3C   CD   7C 0000F       CLRQ      INQUIRE                                          ;
                             FF44   CD   7C 00013       CLRQ      INQUIRE                                          ;
                             FF4C   CD   7C 00017       CLRQ      INQUIRE                                          ;
                             FF54   CD   7C 0001B       CLRQ      INQUIRE                                          ;
                             FF5C   CD   7C 0001F       CLRQ      INQUIRE                                          ;
                             FF64   CD   7C 00023       CLRQ      INQUIRE                                          ;
                             FF6C   CD   7C 00027       CLRQ      INQUIRE                                          ;
                             FF74   CD   7C 0002B       CLRQ      INQUIRE                                          ;
                             FF7C   CD   7C 0002F       CLRQ      INQUIRE                                          ;
                               84   AD   7C 00033       CLRQ      INQUIRE                                          ;
                               8C   AD   7C 00036       CLRQ      INQUIRE                                          ;
                               94   AD   7C 00039       CLRQ      INQUIRE                                          ;
                               9C   AD   7C 0003C       CLRQ      INQUIRE                                          ;
                               A4   AD   7C 0003F       CLRQ      INQUIRE                                          ;
                               AC   AD   7C 00042       CLRQ      INQUIRE                                          ;
                               B4   AD   7C 00045       CLRQ      INQUIRE                                          ;
                               BC   AD   7C 00048       CLRQ      INQUIRE                                          ;
                               C4   AD   7C 0004B       CLRQ      INQUIRE                                          ;
                               CC   AD   7C 0004E       CLRQ      INQUIRE                                          ;
                               D4   AD   7C 00051       CLRQ      INQUIRE                                          ;
                               DC   AD   7C 00054       CLRQ      INQUIRE                                          ;
                               E4   AD   7C 00057       CLRQ      INQUIRE                                          ;
                               EC   AD   D4 0005A       CLRL      L_UNWIND_ACTION                                  ;
                       6D    05AE   CF   DE 0005D       MOVAL     87$, (FP)                                        ;
```

```
    0050  8F          00               6E           00 2C 00062      MOVC5   #0, (SP), #0, #80, $RMS_PTR                      ; 0390
                                            FE7C  CD    00069                                                               ;
                                FE7C   CD     5003  8F B0 0006C      MOVW    #20483, $RMS_PTR                                 ;
                                FE92   CD     4F02  8F B0 00073      MOVW    #20226, $RMS_PTR+22                              ;
                                FE9B   CD           02 90 0007A      MOVB    #2, $RMS_PTR+31                                  ;
                                FEA0   CD     FE50  CD 9E 0007F      MOVAB   XAB_BLOCK, $RMS_PTR+36                           ;
                                FEA4   CD     FECC  CD 9E 00086      MOVAB   NAM_BLOCK, $RMS_PTR+40                           ;
                                FEAC   CD     FF6B  CF 9E 0008D      MOVAB   P.AAW, $RMS_PTR+48                               ;
                                FEB1   CD           04 90 00094      MOVB    #4, $RMS_PTR+53                                  ;
    0060  8F          00               6E           00 2C 00099      MOVC5   #0, (SP), #0, #96, $RMS_PTR                      ; 0392
                                            FECC  CD    000A0                                                               ;
                                FECC   CD     6002  8F B0 000A3      MOVW    #24578, $RMS_PTR                                 ;
                                FECE   CD           01 8E 000AA      MNEGB   #1, $RMS_PTR+2                                   ;
                                FED0   CD     00F8  CE 9E 000AF      MOVAB   RES_OR_EXP_NAME, $RMS_PTR+4                      ;
                                FED6   CD           01 8E 000B6      MNEGB   #1, $RMS_PTR+10                                  ;
                                FED8   CD     00F8  CE 9E 000BB      MOVAB   RES_OR_EXP_NAME, $RMS_PTR+12                     ;
          2C          00               6E           00 2C 000C2      MOVC5   #0, (SP), #0, #44, $RMS_PTR                      ; 0393
                                            FE50  CD    000C7                                                               ;
                                FE50   CD     2C1D  8F B0 000CA      MOVW    #11293, $RMS_PTR                                 ;
    00C0  8F          00               6E           00 2C 000D1      MOVC5   #0, (SP), #0, #192, RESP_VEC                     ; 0398
                                              08  AE    000D8                                                               ;
          30          00               6E           00 2C 000DA      MOVC5   #0, (SP), #0, #48, VAR_LENGTHS                   ; 0399
                                            00C8  CE    000DF                                                               ;
                                     57           3F 8A 000E2      BICB2   #63, INQ_FLAGS                                   ; 0405
                                     5A           01 D0 000E5      MOVL    #1, RET_STATUS                                   ; 0406
                                                  55 D4 000E8      CLRL    RES_OR_EXP_LEN                                   ; 0407
                                  EC   AD           01 D0 000EA      MOVL    #1, L_UNWIND_ACTION                              ; 0414
                                            00C8  CE 9F 000EE      PUSHAB  VAR_LENGTHS                                      ; 0423
                                                  7E D4 000F2      CLRL    -(SP)                                            ;
                                              F0  AD 9F 000F4      PUSHAB  DEF_DSC                                          ;
                                              F8  AD 9F 000F7      PUSHAB  NAM_DSC                                          ;
                                                  2F DD 000FA      PUSHL   #47                                              ;
                                            FF2C  CD 9F 000FC      PUSHAB  INQUIRE                                          ;
                                     7E           6C 9A 00100      MOVZBL  (AP), -(SP)                                      ;
                                              04  AC 9F 00103      PUSHAB  KEYWD                                            ;
                           00000000G   00           08 FB 00106      CALLS   #8, FOR$$OPECLO_ARG                              ;
                                              84  AD D5 0010D      TSTL    INQUIRE+88                                       ; 0432
                                                  10 13 00110      BEQL    2$                                               ;
                                     07       90  AD E9 00112      BLBC    INQUIRE+100, 1$                                  ; 0434
                                00DE   CE           20 90 00116      MOVB    #32, VAR_LENGTHS+22                              ; 0436
                                                  05 11 0011B      BRB     2$                                               ;
                                00DE   CE           10 90 0011D 1$:  MOVB    #16, VAR_LENGTHS+22                              ; 0438
                                     56     FE7C  CD 9E 00122 2$:  MOVAB   FAB_BLOCK, FAB                                   ; 0444
                                     54     FECC  CD 9E 00127      MOVAB   NAM_BLOCK, NAM                                   ; 0445
                                            FF64  CD D5 0012C      TSTL    INQUIRE+56                                       ; 0453
                                                  08 12 00130      BNEQ    3$                                               ;
                                              94  AD D5 00132      TSTL    INQUIRE+104                                      ;
                                                  03 12 00135      BNEQ    3$                                               ;
                                                  016F 31 00137      BRW     24$                                              ;
                                     57           22 88 0013A 3$:  BISB2   #34, INQ_FLAGS                                   ; 0465
                                              94  AD D5 0013D      TSTL    INQUIRE+104                                      ; 0467
                                                  19 13 00140      BEQL    5$                                               ;
                                     50       94  AD D0 00142      MOVL    INQUIRE+104, DNAME                               ; 0472
                                00FF   8F           60 B1 00146      CMPW    (DNAME), #255                                    ; 0473
                                                  0B 1A 0014B      BGTRU   4$                                               ;
                                  35   A6           60 90 0014D      MOVB    (DNAME), 53(FAB)                                 ; 0476
                                  30   A6       04  A0 D0 00151      MOVL    4(DNAME), 48(FAB)                                ; 0477
```

```
                                   03  11 00156        BRB     5$                              ; 0473
                57                 02  8A 00158 4$:    BICB2   #2, INQ_FLAGS                   ; 0480
                         FF64      CD  D5 0015B 5$:    TSTL    INQUIRE+56                      ; 0483
                                   1A  13 0015F        BEQL    7$                              ;
                50       FF64      CD  D0 00161        MOVL    INQUIRE+56, FNAME               ; 0488
          00FF  8F                 60  B1 00166        CMPW    (FNAME), #255                   ; 0489
                                   0B  1A 0016B        BGTRU   6$                              ;
            34  A6                 60  90 0016D        MOVB    (FNAME), 52(FAB)                ; 0492
            2C  A6       04        A0  D0 00171        MOVL    4(FNAME), 44(FAB)               ; 0493
                                   03  11 00176        BRB     7$                              ; 0489
                57                 02  8A 00178 6$:    BICB2   #2, INQ_FLAGS                   ; 0496
     44         57                 01  E1 0017B 7$:    BBC     #1, INQ_FLAGS, 12$              ; 0502
                                   56  DD 0017F        PUSHL   FAB                             ; 0504
      00000000G 00                 01  FB 00181        CALLS   #1, SYS$PARSE                   ;
                21                 50  E9 00188        BLBC    R0, 9$                          ;
      00100038  8F       34        A4  D3 0018B        BITL    52(NAM), #1048632               ; 0511
                                   17  12 00193        BNEQ    9$                              ;
     0B     41  A6                 06  E1 00195        BBC     #6, 65(FAB), 8$                 ; 0519
                                   56  DD 0019A        PUSHL   FAB                             ; 0521
      00000000G 00                 01  FB 0019C        CALLS   #1, SYS$SEARCH                  ;
                                   0A  11 001A3        BRB     10$                             ;
            03  A4       0B        A4  90 001A5 8$:    MOVB    11(NAM), 3(NAM)                 ; 0523
                                   03  11 001AA        BRB     10$                             ; 0504
                57                 02  8A 001AC 9$:    BICB2   #2, INQ_FLAGS                   ; 0527
     10         57                 01  E1 001AF 10$:   BBC     #1, INQ_FLAGS, 12$              ; 0530
                         0B        A4  95 001B3        TSTB    11(NAM)                         ; 0533
                                   03  13 001B6        BEQL    11$                             ;
                57                 02  88 001B8        BISB2   #2, INQ_FLAGS                   ; 0535
                         03        A4  95 001BB 11$:   TSTB    3(NAM)                          ; 0536
                                   03  13 001BE        BEQL    12$                             ;
                57                 08  88 001C0        BISB2   #8, INQ_FLAGS                   ; 0538
            07  A6                 01  88 001C3 12$:   BISB2   #1, 7(FAB)                      ; 0541
     1D         57                 03  E1 001C7        BBC     #3, INQ_FLAGS, 14$              ; 0550
                                   56  DD 001CB        PUSHL   FAB                             ; 0552
      00000000G 00                 01  FB 001CD        CALLS   #1, SYS$OPEN                    ;
                0E                 50  E9 001D4        BLBC    R0, 13$                         ;
                                   56  DD 001D7        PUSHL   FAB                             ; 0555
      00000000G 00                 01  FB 001D9        CALLS   #1, SYS$CLOSE                   ;
                57                 10  88 001E0        BISB2   #16, INQ_FLAGS                  ; 0556
                                   03  11 001E3        BRB     14$                             ; 0552
                57                 08  8A 001E5 13$:   BICB2   #8, INQ_FLAGS                   ; 0559
                55       03        A4  9A 001E8 14$:   MOVZBL  3(NAM), RES_OR_EXP_LEN          ; 0565
                                   04  12 001EC        BNEQ    15$                             ; 0566
                55       0B        A4  9A 001EE        MOVZBL  11(NAM), RES_OR_EXP_LEN         ; 0568
     15         57                 01  E1 001F2 15$:   BBC     #1, INQ_FLAGS, 16$              ; 0581
                         34        A6  94 001F6        CLRB    52(FAB)                         ; 0585
                         2C        A6  7C 001F9        CLRQ    44(FAB)                         ; 0584
                         0C        A4  D4 001FC        CLRL    12(NAM)                         ; 0587
                         04        A4  D4 001FF        CLRL    4(NAM)                          ; 0588
                                   56  DD 00202        PUSHL   FAB                             ; 0589
      00000000G 00                 01  FB 00204        CALLS   #1, SYS$PARSE                   ;
     03         57                 01  E0 0020B 16$:   BBS     #1, INQ_FLAGS, 17$              ; 0608
                               008B  31 0020F        BRW     22$                             ;
            04  A4     00F8        CE  9E 00212 17$:   MOVAB   RES_OR_EXP_NAME, 4(NAM)         ;
            03  A4                 55  90 00218        MOVB    RES_OR_EXP_LEN, 3(NAM)          ; 0609
                         04        AE  D4 0021C        CLRL    LUN_FLAG                        ; 0614
                03                 5A  E8 0021F        BLBS    RET_STATUS, 18$                 ; 0615
```

```
                                   0082  31 00222        BRW     23$                                   ;
                                     5E  DD 00225  18$:  PUSHL   SP                                    ; 0618
                             08      AE  9F 00227        PUSHAB  LUN_FLAG                              ;
              00000000G  00          02  FB 0022A        CALLS   #2, FOR$$NEXT_LUN                     ;
                             04      AE  D5 00231        TSTL    LUN_FLAG                              ; 0619
                                     60  13 00234        BEQL    21$                                   ;
                                     7E  D4 00236        CLRL    -(SP)                                 ; 0632
              00000000G  00          01  FB 00238        CALLS   #1, SYS$SETAST                        ;
                         58          50  D0 0023F        MOVL    R0, AST_STATUS                        ;
                                     6E  DD 00242        PUSHL   UNIT                                  ; 0633
              00000000G  00          01  FB 00244        CALLS   #1, FOR$$CB_FETCH                     ;
                                     5B  D5 0024B        TSTL    CCB                                   ; 0634
                                     39  13 0024D        BEQL    20$                                   ;
                         35  FC      AB  E9 0024F        BLBC    -4(CCB), 20$                          ; 0636
                         50  F7      AB  9A 00253        MOVZBL  -9(CCB), R0                           ; 0638
   55          20    F8  BB          50  2D 00257        CMPC5   R0, @-8(CCB), #32, RES_OR_EXP_LEN, -  ;
                             00F8    CE     0025D                RES_OR_EXP_NAME                       ;
                                     26  12 00260        BNEQ    20$                                   ;
                         57          01  88 00262        BISB2   #1, INQ_FLAGS                         ; 0647
                                     6E  DD 00265        PUSHL   UNIT                                  ; 0648
                  0000V  CF          01  FB 00267        CALLS   #1, PUSH_CCB                          ;
                         59          50  D0 0026C        MOVL    R0, STATUS                            ;
                         09          58  D1 0026F        CMPL    AST_STATUS, #9                        ; 0649
                                     09  12 00272        BNEQ    19$                                   ;
                                     01  DD 00274        PUSHL   #1                                    ; 0651
              00000000G  00          01  FB 00276        CALLS   #1, SYS$SETAST                        ;
                         67          59  E9 0027D  19$:  BLBC    STATUS, 27$                           ; 0652
                             EC      AD  D4 00280        CLRL    L_UNWIND_ACTION                       ; 0655
                         57          0C  88 00283        BISB2   #12, INQ_FLAGS                        ; 0657
                                     5F  11 00286        BRB     27$                                   ; 0654
                         09          58  D1 00288  20$:  CMPL    AST_STATUS, #9                        ; 0672
                                     09  12 0028B        BNEQ    21$                                   ;
                                     01  DD 0028D        PUSHL   #1                                    ; 0674
              00000000G  00          01  FB 0028F        CALLS   #1, SYS$SETAST                        ;
                             04      AE  D5 00296  21$:  TSTL    LUN_FLAG                              ; 0677
                                     8A  12 00299        BNEQ    18$                                   ;
                                     4A  11 0029B        BRB     27$                                   ; 0616
                     04  A4  2C      A6  D0 0029D  22$:  MOVL    44(FAB), 4(NAM)                       ; 0686
                     03  A4  34      A6  90 002A2        MOVB    52(FAB), 3(NAM)                       ; 0687
                                     3E  11 002A7  23$:  BRB     27$                                   ; 0453
                         6E  FF30    CD  D0 002A9  24$:  MOVL    INQUIRE+4, UNIT                       ; 0698
                         50          6E  D0 002AE        MOVL    UNIT, R0                              ; 0699
                                     34  19 002B1        BLSS    27$                                   ;
               00000077  8F          50  D1 002B3        CMPL    R0, #119                              ;
                                     2B  14 002BA        BGTR    27$                                   ;
                         57          01  88 002BC        BISB2   #1, INQ_FLAGS                         ; 0702
                                     50  DD 002BF        PUSHL   R0                                    ; 0710
                  0000V  CF          01  FB 002C1        CALLS   #1, PUSH_CCB                          ;
                         59          50  D0 002C6        MOVL    R0, STATUS                            ;
                         18          59  E9 002C9        BLBC    STATUS, 26$                           ; 0711
                             EC      AD  D4 002CC        CLRL    L_UNWIND_ACTION                       ; 0717
                         05  FC      AB  E9 002CF        BLBC    -4(CCB), 25$                          ; 0718
                         57          0E  88 002D3        BISB2   #14, INQ_FLAGS                        ; 0723
                                     0F  11 002D6        BRB     27$                                   ; 0718
                       00000000G     00  16 002D8  25$:  JSB     FOR$$CB_POP                           ; 0730
                     EC  AD          01  D0 002DE        MOVL    #1, L_UNWIND_ACTION                   ; 0731
                                     03  11 002E2        BRB     27$                                   ; 0711
```

```
                                5A            59 D0 002E4 26$:   MOVL    STATUS, RET_STATUS                             ; 0738
                   09           57            02 E1 002E7 27$:   BBC     #2, INQ_FLAGS, 28$                             ; 0747
                                56       44   AB 9E 002EB        MOVAB   68(R11), FAB                                   ; 0750
                                54     0094   CB 9E 002EF        MOVAB   148(R11), NAM                                  ; 0751
                                         A4   AD D5 002F4 28$:   TSTL    INQUIRE+120                                    ; 0763
                                              16 13 002F7        BEQL    31$                                            ;
                   07           57            03 E0 002F9        BBS     #3, INQ_FLAGS, 29$                             ; 0765
                   0A           57            05 E0 002FD        BBS     #5, INQ_FLAGS, 30$                             ;
                                07            57 E9 00301        BLBC    INQ_FLAGS, 30$                                 ; 0766
                         0080   CE            01 CE 00304 29$:   MNEGL   #1, RESP_VEC+120                               ; 0768
                                              04 11 00309        BRB     31$                                            ;
                                       0080   CE D4 0030B 30$:   CLRL    RESP_VEC+120                                   ; 0770
                                         A8   AD D5 0030F 31$:   TSTL    INQUIRE+124                                    ; 0776
                                              16 13 00312        BEQL    34$                                            ;
                   07           57            02 E0 00314        BBS     #2, INQ_FLAGS, 32$                             ; 0778
                   0A           57            05 E1 00318        BBC     #5, INQ_FLAGS, 33$                             ; 0779
                                07            57 E9 0031C        BLBC    INQ_FLAGS, 33$                                 ;
                         0084   CE            01 CE 0031F 32$:   MNEGL   #1, RESP_VEC+124                               ; 0781
                                              04 11 00324        BRB     34$                                            ;
                                       0084   CE D4 00326 33$:   CLRL    RESP_VEC+124                                   ; 0783
                                         AC   AD D5 0032A 34$:   TSTL    INQUIRE+128                                    ; 0791
                                              0E 13 0032D        BEQL    36$                                            ;
                                07            57 E9 0032F        BLBC    INQ_FLAGS, 35$                                 ; 0793
                         0088   CE            6E D0 00332        MOVL    UNIT, RESP_VEC+128                             ; 0795
                                              04 11 00337        BRB     36$                                            ;
                                       0088   CE D4 00339 35$:   CLRL    RESP_VEC+128                                   ; 0797
                                         B0   AD D5 0033D 36$:   TSTL    INQUIRE+132                                    ; 0804
                                              1C 13 00340        BEQL    39$                                            ;
                   07           57            01 E1 00342        BBC     #1, INQ_FLAGS, 37$                             ; 0807
                         008C   CE            01 CE 00346        MNEGL   #1, RESP_VEC+132                               ; 0809
                                              04 11 0034B        BRB     38$                                            ;
                                       008C   CE D4 0034D 37$:   CLRL    RESP_VEC+132                                   ; 0811
                   09           57            02 E1 00351 38$:   BBC     #2, INQ_FLAGS, 39$                             ; 0812
                   04      FC   AB            05 E1 00355        BBC     #5, -4(CCB), 39$                               ;
                                       008C   CE D4 0035A        CLRL    RESP_VEC+132                                   ; 0814
                                         B4   AD D5 0035E 39$:   TSTL    INQUIRE+136                                    ; 0822
                                              10 13 00361        BEQL    40$                                            ;
                                50       B4   AD D0 00363        MOVL    INQUIRE+136, NAME_DSC                          ; 0827
                                51       03   A4 9A 00367        MOVZBL  3(NAM), R1                                     ; 0828
         60        20      04   B4            51 2C 0036B        MOVC5   R1, @4(NAM), #32, (NAME_DSC), @4(NAME_DSC)     ; 0829
                                         04   B0    00371                                                               ;
                                         B8   AD D5 00373 40$:   TSTL    INQUIRE+140                                    ; 0837
                                              2D 13 00376        BEQL    44$                                            ;
                         0094   CE            03 D0 00378        MOVL    #3, RESP_VEC+140                               ; 0840
                   1F           57            02 E1 0037D        BBC     #2, INQ_FLAGS, 43$                             ; 0841
                   07      FC   AB            04 E1 00381        BBC     #4, -4(CCB), 41$                               ; 0843
                         0094   CE            04 D0 00386        MOVL    #4, RESP_VEC+140                               ; 0845
                                              18 11 0038B        BRB     44$                                            ;
                                         FD   AB 95 0038D 41$:   TSTB    -3(CCB)                                        ; 0846
                                              07 18 00390        BGEQ    42$                                            ;
                         0094   CE            05 D0 00392        MOVL    #5, RESP_VEC+140                               ; 0848
                                              0C 11 00397        BRB     44$                                            ;
                         0094   CE            06 D0 00399 42$:   MOVL    #6, RESP_VEC+140                               ; 0850
                                              05 11 0039E        BRB     44$                                            ; 0843
                         0094   CE            03 D0 003A0 43$:   MOVL    #3, RESP_VEC+140                               ; 0852
                                         BC   AD D5 003A5 44$:   TSTL    INQUIRE+144                                    ; 0861
                                              10 13 003A8        BEQL    46$                                            ;
```

```
                07              57              03  E1  003AA          BBC     #3, INQ_FLAGS, 45$                      ; 0863
                        0098    CE              01  D0  003AE          MOVL    #1, RESP_VEC+144                        ; 0865
                                                05  11  003B3          BRB     46$
                        0098    CE              03  D0  003B5  45$:     MOVL    #3, RESP_VEC+144                        ; 0867
                            C0  AD              D5  003BA  46$:     TSTL    INQUIRE+T48                             ; 0874
                                                2D  13  003BD          BEQL    51$
                        009C    CE              03  D0  003BF          MOVL    #3, RESP_VEC+148                        ; 0877
                24              57              03  E1  003C4          BBC     #3, INQ_FLAGS, 51$                      ; 0878
                                50          1D  A6  9A  003C8          MOVZBL  29(FAB), R0                             ; 0881
                                                08  12  003CC          BNEQ    47$                                     ; 0884
                                01          1F  A6  91  003CE          CMPB    31(FAB), #1                             ; 0885
                                                13  12  003D2          BNEQ    50$
                                                05  11  003D4          BRB     48$                                     ; 0887
                                10              50  91  003D6  47$:     CMPB    R0, #16                                 ; 0890
                                                07  12  003D9          BNEQ    49$
                        009C    CE              01  D0  003DB  48$:     MOVL    #1, RESP_VEC+148                        ; 0891
                                                0A  11  003E0          BRB     51$
                                20              50  91  003E2  49$:     CMPB    R0, #32                                 ; 0892
                                                05  12  003E5          BNEQ    51$
                        009C    CE              02  D0  003E7  50$:     MOVL    #2, RESP_VEC+148                        ; 0893
                            E4  AD              D5  003EC  51$:     TSTL    INQUIRE+T84                             ; 0904
                                                27  13  003EF          BEQL    54$
                        00C0    CE              03  D0  003F1          MOVL    #3, RESP_VEC+184                        ; 0907
                1E              57              03  E1  003F6          BBC     #3, INQ_FLAGS, 54$                      ; 0908
                                50          1D  A6  9A  003FA          MOVZBL  29(FAB), R0                             ; 0911
                                20              50  91  003FE          CMPB    R0, #32                                 ; 0914
                                                07  12  00401          BNEQ    52$
                        00C0    CE              01  D0  00403          MOVL    #1, RESP_VEC+184                        ; 0915
                                                0E  11  00408          BRB     54$
                                                50  D5  0040A  52$:     TSTL    R0                                      ; 0916
                                                05  13  0040C          BEQL    53$
                                10              50  91  0040E          CMPB    R0, #16
                                                05  12  00411          BNEQ    54$
                        00C0    CE              02  D0  00413  53$:     MOVL    #2, RESP_VEC+184                        ; 0917
                            C4  AD              D5  00418  54$:     TSTL    INQUIRE+T52                             ; 0930
                                                1E  13  0041B          BEQL    56$
                        00A0    CE              03  D0  0041D          MOVL    #3, RESP_VEC+152                        ; 0933
                15              57              02  E1  00422          BBC     #2, INQ_FLAGS, 56$                      ; 0934
                                07          FD  AB  E9  00426          BLBC    -3(CCB), 55$                            ; 0936
                        00A0    CE              07  D0  0042A          MOVL    #7, RESP_VEC+152                        ; 0938
                                                0A  11  0042F          BRB     56$
                05          FD  AB              01  E1  00431  55$:     BBC     #1, -3(CCB), 56$                        ; 0939
                        00A0    CE              08  D0  00436          MOVL    #8, RESP_VEC+152                        ; 0941
                            C8  AD              D5  0043B  56$:     TSTL    INQUIRE+T56                             ; 0951
                                                10  13  0043E          BEQL    58$
                07              57              03  E1  00440          BBC     #3, INQ_FLAGS, 57$                      ; 0953
                        00A4    CE              01  D0  00444          MOVL    #1, RESP_VEC+156                        ; 0955
                                                05  11  00449          BRB     58$
                        00A4    CE              03  D0  0044B  57$:     MOVL    #3, RESP_VEC+156                        ; 0957
                            CC  AD              D5  00450  58$:     TSTL    INQUIRE+T60                             ; 0966
                                                10  13  00453          BEQL    60$
                07              57              03  E1  00455          BBC     #3, INQ_FLAGS, 59$                      ; 0968
                        00A8    CE              01  D0  00459          MOVL    #1, RESP_VEC+160                        ; 0970
                                                05  11  0045E          BRB     60$
                        00A8    CE              03  D0  00460  59$:     MOVL    #3, RESP_VEC+160                        ; 0972
                            DC  AD              D5  00465  60$:     TSTL    INQUIRE+T76                             ; 0980
                                                2C  13  00468          BEQL    63$
```

```
                 00B8   CE          03  D0 0046A        MOVL    #3, RESP_VEC+176                    ; 0983
     23                 57          03  E1 0046F        BBC     #3, INQ_FLAGS, 63$                  ; 0984
                        50    1D    A6  9A 00473        MOVZBL  29(FAB), R0                         ; 0986
                                    07  12 00477        BNEQ    61$                                 ; 0989
                 00B8   CE          06  D0 00479        MOVL    #6, RESP_VEC+176                    ; 0990
                                    16  11 0047E        BRB     63$                                 ;
                        10          50  91 00480 61$:   CMPB    R0, #16                             ; 0991
                                    07  12 00483        BNEQ    62$                                 ;
                 00B8   CE          09  D0 00485        MOVL    #9, RESP_VEC+176                    ; 0992
                                    0A  11 0048A        BRB     63$                                 ;
                        20          50  91 0048C 62$:   CMPB    R0, #32                             ; 0993
                                    05  12 0048F        BNEQ    63$                                 ;
                 00B8   CE          0A  D0 00491        MOVL    #10, RESP_VEC+176                   ; 0994
                              D0    AD  D5 00496 63$:   TSTL    INQUIRE+164                         ; 1011
                                    42  13 00499        BEQL    67$                                 ;
                            00AC    CE  D4 0049B        CLRL    RESP_VEC+164                        ; 1014
     21                 57          02  E1 0049F        BBC     #2, INQ_FLAGS, 65$                  ; 1015
                              D2    AB  B5 004A3        TSTW    -46(CCB)                            ; 1018
                                    35  13 004A6        BEQL    67$                                 ;
                 00AC   CE    D2    AB  3C 004A8        MOVZWL  -46(CCB), RESP_VEC+164              ; 1021
     05            FD   AB          03  E1 004AE        BBC     #3, -3(CCB), 64$                    ; 1022
                 00AC   CE          02  C2 004B3        SUBL2   #2, RESP_VEC+164                    ; 1024
     20            FD   AB          01  E1 004B8 64$:   BBC     #1, -3(CCB), 67$                    ; 1025
                 00AC   CE          04  C6 004BD        DIVL2   #4, RESP_VEC+164                    ; 1027
                                    19  11 004C2        BRB     67$                                 ; 1015
     15                 57          04  E1 004C4 65$:   BBC     #4, INQ_FLAGS, 67$                  ; 1030
                        50    36    A6  3C 004C8        MOVZWL  54(FAB), R0                         ; 1033
                        50  FE5A    CD  B1 004CC        CMPW    XAB_BLOCK+10, R0                    ;
                                    05  1B 004D1        BLEQU   66$                                 ;
                        50  FE5A    CD  3C 004D3        MOVZWL  XAB_BLOCK+10, R0                    ;
                 00AC   CE          50  D0 004D8 66$:   MOVL    R0, RESP_VEC+164                    ; 1032
                              D4    AD  D5 004DD 67$:   TSTL    INQUIRE+168                         ; 1041
                                    13  13 004E0        BEQL    68$                                 ;
                            00B0    CE  D4 004E2        CLRL    RESP_VEC+168                        ; 1044
     0B                 57          02  E1 004E6        BBC     #2, INQ_FLAGS, 68$                  ; 1045
     06            FC   AB          04  E1 004EA        BBC     #4, -4(CCB), 68$                    ;
                 00B0   CE    E0    AB  D0 004EF        MOVL    -32(CCB), RESP_VEC+168              ; 1047
                              D8    AD  D5 004F5 68$:   TSTL    INQUIRE+172                         ; 1056
                                    1E  13 004F8        BEQL    70$                                 ;
                 00B4   CE          03  D0 004FA        MOVL    #3, RESP_VEC+172                    ; 1059
     15                 57          02  E1 004FF        BBC     #2, INQ_FLAGS, 70$                  ; 1060
                        11    FD    AB  E9 00503        BLBC    -3(CCB), 70$                        ;
     07            FF   AB          06  E1 00507        BBC     #6, -1(CCB), 69$                    ; 1063
                 00B4   CE          0B  D0 0050C        MOVL    #11, RESP_VEC+172                   ; 1065
                                    05  11 00511        BRB     70$                                 ;
                 00B4   CE          0C  D0 00513 69$:   MOVL    #12, RESP_VEC+172                   ; 1067
                              E0    AD  D5 00518 70$:   TSTL    INQUIRE+180                         ; 1077
                                    5A  13 0051B        BEQL    76$                                 ;
                 00BC   CE          03  D0 0051D        MOVL    #3, RESP_VEC+180                    ; 1080
     43                 57          03  E1 00522        BBC     #3, INQ_FLAGS, 75$                  ; 1081
                        50    1F    A6  9A 00526        MOVZBL  31(FAB), R0                         ; 1083
                        01          50  91 0052A        CMPB    R0, #1                              ; 1086
                                    07  12 0052D        BNEQ    71$                                 ;
                 00BC   CE          0D  D0 0052F        MOVL    #13, RESP_VEC+180                   ; 1087
                                    33  11 00534        BRB     75$                                 ;
                        02          50  91 00536 71$:   CMPB    R0, #2                              ; 1088
                                    0C  1F 00539        BLSSU   72$                                 ;
```

```
                03        50 91 0053B        CMPB    R0, #3                                      ;
                          07 1A 0053E        BGTRU   72$                                         ;
         00BC   CE        0E D0 00540        MOVL    #14, RESP_VEC+180                           ; 1089
                          22 11 00545        BRB     75$                                         ;
                04        50 91 00547 72$:   CMPB    R0, #4                                      ; 1090
                          07 12 0054A        BNEQ    73$                                         ;
         00BC   CE        13 D0 0054C        MOVL    #19, RESP_VEC+180                           ; 1091
                          16 11 00551        BRB     75$                                         ;
                06        50 91 00553 73$:   CMPB    R0, #6                                      ; 1092
                          07 12 00556        BNEQ    74$                                         ;
         00BC   CE        14 D0 00558        MOVL    #20, RESP_VEC+180                           ; 1093
                          0A 11 0055D        BRB     75$                                         ;
                05        50 91 0055F 74$:   CMPB    R0, #5                                      ; 1094
                          05 12 00562        BNEQ    75$                                         ;
         00BC   CE        15 D0 00564        MOVL    #21, RESP_VEC+180                           ; 1095
     0A         57        02 E1 00569 75$:   BBC     #2, INQ_FLAGS, 76$                          ; 1097
     05    FD   AB        03 E1 0056D        BBC     #3, -3(CCB), 76$                            ; 1099
         00BC   CE        0F D0 00572        MOVL    #15, RESP_VEC+180                           ; 1101
                     E8   AD D5 00577 76$:   TSTL    INQUIRE+188                                 ; 1107
                          2A 13 0057A        BEQL    79$                                         ;
         00C4   CE        03 D0 0057C        MOVL    #3, RESP_VEC+188                            ; 1110
     21         57        03 E1 00581        BBC     #3, INQ_FLAGS, 79$                          ; 1111
                07   1E   A6 E9 00585        BLBC    30(FAB), 77$                                ; 1114
         00C4   CE        10 D0 00589        MOVL    #16, RESP_VEC+188                           ; 1116
                          16 11 0058E        BRB     79$                                         ;
     07    1E   A6        01 E1 00590 77$:   BBC     #1, 30(FAB), 78$                            ; 1117
         00C4   CE        11 D0 00595        MOVL    #17, RESP_VEC+188                           ; 1119
                          0A 11 0059A        BRB     79$                                         ;
     05    1E   A6        02 E0 0059C 78$:   BBS     #2, 30(FAB), 79$                            ; 1120
         00C4   CE        12 D0 005A1        MOVL    #18, RESP_VEC+188                           ; 1122
                58        16 D0 005A6 79$:   MOVL    #22, I                                      ; 1134
                50   FF2C CD48 DE 005A9 80$: MOVAL   INQUIRE[I], R0                              ; 1135
                          60 D5 005AF        TSTL    (R0)                                        ;
                          3C 13 005B1        BEQL    84$                                         ;
     02         00   FA14 CF48 8F 005B3      CASEB   RESP_TYPES[I], #0, #2                       ; 1137
   0017       0008        0035 005BA 81$:    .WORD   84$-81$,-                                   ;
                                                     82$-81$,-                                   ;
                                                     83$-81$                                     ;
                          2D 11 005C0        BRB     84$                                         ;
                51        60 D0 005C2 82$:   MOVL    (R0), DEST                                  ; 1147
 61   00C8 CE48 00   08 AE48 F0 005C5        INSV    RESP_VEC[I], #0, VAR_LENGTHS[I], (DEST)     ; 1148
                          1E 11 005CF        BRB     84$                                         ; 1137
                51        60 D0 005D1 83$:   MOVL    (R0), DSC                                   ; 1156
                50   08 AE48 D0 005D4        MOVL    RESP_VEC[I], WHICH                          ; 1157
                52   F9D6 CF40 9A 005D9      MOVZBL  RESP_LENS[WHICH], R2                        ; 1158
                50   F978 CF40 D0 005DF      MOVL    RESP_VALS[WHICH], R0                        ;
 61        20   F971 CF40 52 2C 005E5        MOVC5   R2, RESP_VALS[R0], #32, (DSC), @4(DSC)      ; 1159
                     04   B1    005ED                                                            ;
     B6         58        2F F3 005EF 84$:   AOBLEQ  #47, I, 80$                                 ; 1135
                0B        5A E8 005F3        BLBS    RET_STATUS, 85$                             ; 1166
                          6E DD 005F6        PUSHL   UNIT                                        ; 1168
                          5A DD 005F8        PUSHL   RET_STATUS                                  ;
      00000000G 00        02 FB 005FA        CALLS   #2, FOR$$SIG_NO_LUB                         ;
     06         57        02 E1 00601 85$:   BBC     #2, INQ_FLAGS, 86$                          ; 1173
           00000000G 00   16 00605           JSB     FOR$$CB_POP                                 ; 1175
                50        01 D0 0060B 86$:   MOVL    #1, R0                                      ; 1182
                             04 0060E        RET                                                 ; 1183
```

```
                              0000 0060F 87$:   .WORD   Save nothing                    ; 0347
          50       08    AC   D0 00611          MOVL    8(AP), R0                       ;
          50       04    A0   D0 00615          MOVL    4(R0), R0                       ;
                 FF2C    C0   9F 00619          PUSHAB  INQUIRE                         ;
                   EC    A0   9F 0061D          PUSHAB  L_UNWIND_ACTION                 ;
                         02   DD 00620          PUSHL   #2                              ;
                         5E   DD 00622          PUSHL   SP                              ;
          7E       04    AC   7D 00624          MOVQ    4(AP), -(SP)                    ;
 00000000G 00            03   FB 00628          CALLS   #3, FOR$$ERR_OPECLO             ;
                              04 0062F          RET                                     ;

; Routine Size:  1584 bytes,    Routine Base:  _FOR$CODE + 0130
```

```
1 ROUTINE PUSH_CCB (                    ! Call FOR$$CB_PUSH and return with value
1       UNIT) : CALL_CCB =
1
1 !++
1 ! ABSTRACT:
1 !
1 !       Call FOR$$CB_PUSH and return with a condition value as the function
1 !       result.  FOR$$CB_PUSH is not called directly as it may signal when
1 !       we do not desire it.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       UNIT    - The unit to be pushed
1 !
1 ! FUNCTION RESULT:
1 !
1 !       Either SS$_NORMAL or the condition code of an error which was
1 !       signalled by FOR$$CB_PUSH.
1 !--
1
2 BEGIN
2
2     EXTERNAL REGISTER
2         CCB: REF $FOR$CCB_DECL;
2
2     ENABLE
2         LIB$SIG_TO_RET;                           ! Convert signals to return values
2
2     FOR$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);         ! Push the CCB
2
2     RETURN 1;                                     ! Success
2
1     END;



                                  0004 00000 PUSH_CCB:
                                                      .WORD   Save R2                       ; 1184
          6D       0012  CF  DE 00002                 MOVAL   1$, (FP)                      ; 1204
          50             08  CE 00007                 MNEGL   #8, R0                        ; 1212
          52         04  AC  D0 0000A                 MOVL    UNIT, R2                      ;
            00000000G    00  16 0000E                 JSB     FOR$$CB_PUSH                  ;
          50             01  D0 00014                 MOVL    #1, R0                        ; 1214
                             04 00017                 RET                                   ; 1216
                           0000 00018 1$:             .WORD   Save nothing                  ; 1204
                         7E  D4 0001A                 CLRL    -(SP)                         ;
                         5E  DD 0001C                 PUSHL   SP                            ;
          7E         04  AC  7D 0001E                 MOVQ    4(AP), -(SP)                  ;
   00000000G  00         03  FB 00022                 CALLS   #3, LIB$SIG_TO_RET            ;
                             04 00029                 RET                                   ;

; Routine Size:  42 bytes,    Routine Base:  _FOR$CODE + 0760
```

```
; 1159          1217  1
; 1160          1218  1 END                                     !End of module
; 1161          1219  0 ELUDOM
```

```
;                                PSECT SUMMARY
;
;       Name                     Bytes                          Attributes
;
;  _FOR$CODE                       1930  NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
```

```
;                            Library Statistics
;
;                                          -------- Symbols --------        Pages        Processing
;       File                               Total    Loaded   Percent       Mapped        Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1        9776        96         0          581        00:01.0
;  _$255$DUA28:[FORRTL.OBJ]FORLIB.L32;1      711       212        29           52        00:00.6
;  _$255$DUA28:[FORRTL.OBJ]RTLLIB.L32;1       36         0         0            8        00:00.1
```

```
;                                COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:FORINQUIR/OBJ=OBJ$:FORINQUIR MSRC$:FORINQUIR/UPDATE=(ENH$:FORINQUIR
;       )
;
; Size:           1626 code + 304 data bytes
; Run Time:          00:38.2
; Elapsed Time:      01:44.9
; Lines/CPU Min:      1914
; Lexemes/CPU-Min: 22410
; Memory Used:   500 pages
; Compilation Complete
```