```
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

K 6

```
**FILE**ID**FOROPEN

FFFFFFFFFF    OOOOOO    RRRRRRRR      OOOOOO    PPPPPPPP    EEEEEEEEEE  NN        NN
FFFFFFFFFF    OOOOOO    RRRRRRRR      OOOOOO    PPPPPPPP    EEEEEEEEEE  NN        NN
FF          OO      OO  RR      RR  OO      OO  PP      PP  EE          NN        NN
FF          OO      OO  RR      RR  OO      OO  PP      PP  EE          NN        NN
FF          OO      OO  RR      RR  OO      OO  PP      PP  EE          NNNN      NN
FF          OO      OO  RR      RR  OO      OO  PP      PP  EE          NNNN      NN
FFFFFFFF    OO      OO  RRRRRRRR    OO      OO  PPPPPPPP    EEEEEEEE    NN  NN    NN
FFFFFFFF    OO      OO  RRRRRRRR    OO      OO  PPPPPPPP    EEEEEEEE    NN  NN    NN
FF          OO      OO  RR  RR      OO      OO  PP          EE          NN    NNNN
FF          OO      OO  RR  RR      OO      OO  PP          EE          NN    NNNN
FF          OO      OO  RR    RR    OO      OO  PP          EE          NN        NN    ....
FF          OO      OO  RR    RR    OO      OO  PP          EE          NN        NN    ....
FF            OOOOOO    RR      RR    OOOOOO    PP          EEEEEEEEEE  NN        NN    ....
FF            OOOOOO    RR      RR    OOOOOO    PP          EEEEEEEEEE  NN        NN    ....

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
```

FO
1-

```
0 MODULE FOR$OPEN (%TITLE 'FORTRAN OPEN'
0                 IDENT = '1-065'          ! File: FOROPEN.B32  Edit: SBL1065
0                 ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY: FORTRAN Support Library - user callable
1 !
1 ! ABSTRACT:
1 !
1 !       This module opens a file on a specified logical unit
1 !       (LUN) and allocates 3 control blocks for use by subsequent
1 !       I/O statement calls for this LUN. The 3 control blocks
1 !       are: Logical Unit Block (LUB), I/O statement Block (ISB),
1 !       and an RMS Record Access Block (RAB).
1 !
1 ! ENVIRONMENT: User access mode; mixture of AST level or not.
1 !
1 ! AUTHOR:       Thomas N. Hastings, CREATION DATE: 6-Mar-77; Version 0
1 !
1 ! MODIFIED BY:
1 !
1 !       Thomas N. Hastings, 15-Mar-77: Version 0
1 ! [Previous edit history removed.  SBL 5-Oct-1982]
1 ! 1-062 - Move the BUILTIN ACTUALCOUNT into the routine that needs it, in
1 !         anticipation of the next BLISS compiler, which will require it
1 !         to be there.  While we are here, improve the source text layout.
1 !         Note that this edit changes no code.  JBS 27-Aug-1980
1 ! 1-063 - Add support for DEFAULTFILE keyword.  JAW 30-Jun-1981
1 ! 1-064 - Allow DEFAULTFILE value to be ASCIZ.  JAW 30-Jun-1981
1 ! 1-065 - Reflect separation of FOR$$ data structures from FOR$$.  SBL 5-Oct-1982
1 !--
1
```

```
1 !
1 ! PROLOGUE FILE:
1 !
1
1 REQUIRE 'RTLIN:FORPROLOG';                                   ! FORTRAN Declarations
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     FOR$OPEN,                                                ! FORTRAN OPEN statement
1     FOR$$OPECLO_ARG : NOVALUE,                               ! Get OPEN/CLOSE arguments
1     OPEN_ON_CONNECTED : CALL_CCB;                            ! open on a connected unit
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !       NONE
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 EXTERNAL ROUTINE
1     FOR$$ERR_OPECLO,                                         ! OPEN/CLOSE condition handler
1     FOR$$OPEN_PROC : CALL_CCB NOVALUE,                       ! Does the actual OPEN
1     FOR$$SIGNAL_STO : NOVALUE,                               ! Convert small FORTRAN err #
1                                                              ! to 32-bit VAX error # and SIGNAL_STOP
1     FOR$$SIG_NO_LUB : NOVALUE,                               ! same as FOR$$SIGNAL_STO except no LUB setup
1                                                              ! so must pass LUN explicitly.
1     FOR$$CB_PUSH : JSB_CB_PUSH NOVALUE,                      ! push current LUB/ISB/RAB, if any, and allocate LUB/ISB/RAB
1                                                              ! for this logical unit
1     FOR$$CB_POP : JSB_CB_POP NOVALUE,                        ! Pop I/O system back to previous LUB or indicate
1                                                              ! no I/O statement is currently being processed.
1     FOR$$OPEN_KEYWD,                                         ! Look up keywords for literal values
1     FOR$$SIG_FATINT : NOVALUE,                               ! Signal_stop internal error
1     FOR$$CLOSE_FILE : CALL_CCB;                              ! Close a file
1
```

```
1 GLOBAL ROUTINE FOR$OPEN (                               ! FORTRAN OPEN statement
1           KEYWD,                                        ! keyword code - repeated arguments
1           INFO                                          ! value of keyword (optional)
1     ) =                                                 ! value is TRUE iff successful, FALSE if error and ERR=
1
1 !++
1 ! ABSTRACT:
1 !
1 !         Open file on the specified logical unit (LUN) with
1 !         attributes specified in the keyword parameters and allocate
1 !         3 control blocks for use by subsequent I/O statement calls
1 !         for this LUN.  The 3 control blocks are: Logical Unit
1 !         Block (LUB), I/O statement block (ISB), and one RMS
1 !         control block: the RAB.  If a previous CALL ASSIGN
1 !         or CALL FDBSET has been done all of these control blocks
1 !         have already been allocated, and a FAB has been
1 !         allocated to hold the information passed to CALL ASSIGN or
1 !         CALL FDBSET.
1 !         An RMS $OPEN or $CONNECT is performed.
1 !         Then a record buffer is allocated for the LUN.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !     The following pair is repeated for each user specified keyword:
1 !         KEYWD.rlu.v             Contains KEY<7:0>, ARGTYPE<15:8>, and possibly
1 !                                 INFO<31:16>
1 !         INFO.rlu.v              optional information if need more than
1 !                                 16-bits
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         FOR$$A_CUR_LUB          Current active LUB to be pushed
1 !                                 down or 0 if no LUB has an I/O
1 !                                 statement in progress (usual).
1 !                                 Restored on return from FOR$OPEN
1 !         LUB$V_FAB               1 if FAB allocated by FDBSET, CALL ASSIGN
1 !         LUB$V_DIRECT            1 if DEFINE FILE done
1 !         LUB$V_OPENED            1 if unit already opened
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         LUB$V_READ_ONLY         1 if 'READONLY' present
1 !         LUB$V_DIRECT            1 if ACCESS = 'DIRECT'
1 !         LUB$V_APPEND            1 if ACCESS = 'APPEND'
1 !         LUB$V_OLD_FILE          1 if TYPE = 'OLD'
1 !         LUB$V_SCRATCH           1 if TYPE = 'SCRATCH'
1 !         LUB$V_PRINT             1 if DISPOSE = 'PRINT'
1 !         LUB$V_FIXED             1 if RECORDTYPE = 'FIXED'
1 !         LUB$V_FORMATTED         1 if FORM = 'FORMATTED' or ommitted
1 !         LUB$V_UNFORMAT          1 if FORM = 'UNFORMATTED'
1 !         LUB$A_ASSOC_VAR         adr. of n if ASSOCIATEVARIABLE = n is present
1 !         LUB$V_ASS_VAR_L         1 if n is longword
1 !         LUB$W_IFI               RMS internal file id. Needed in case
1 !                                 FORTRAN CLOSE done.
1 !         LUB$W_RBUF_SIZE         Size in bytes of record buffer allocated.
1 !
1 ! COMPLETTION STATUS:
```

```
!
!       TRUE if success, FALSE if failure and ERR= keyword present
!
! SIDE EFFECTS:
!
!       Allocates LUB/ISB/RAB if not already allocated
!       by CALL ASSIGN, DEFINE FILE, OR CALL FDBSET.
!       SIGNALs or SIGNAL_STOPs the following errors unless ERR=
!       keyword is present: SIGNAL_STOPs FOR$_INCOPECLO (46 =
!       'INCONSISTENT OPEN/CLOSE STATEMENT SPECIFICATIONS')
!       SIGNAL_STOPS FOR$RECIO_OPE (40='RECURSIVE I/O OPERATION')
!       SIGNAL_STOPS FOR$_INVLOGUNI (32='INVALID LOGICAL UNIT NUMBER')
!       See FOR$$OPEN_PROC for other SIGNAL_STOPS.
!
!--

    BEGIN

    GLOBAL REGISTER
        CCB = K_CCB_REG : REF $FOR$CCB_DECL;

!+
! Use the formal arg list as a VECTOR of blocks; each block = 1 longword.
!-

    MAP
        KEYWD : BLOCKVECTOR [255, 1];

    BUILTIN
        ACTUALCOUNT;

    LOCAL
        NAM_DSC : DSC$DESCRIPTOR,                       ! String descriptor for ASCIZ filename
        DEF_DSC : DSC$DESCRIPTOR,                       ! String descriptor for ASCIZ default file name
        L_UNWIND_ACTION : VOLATILE,                     ! UNWIND action code for handler
        OPEN : VOLATILE VECTOR [OPEN$K_KEY_MAX + 1];    ! open parameter array

!+
! Establish handler to RESIGNAL or UNWIND if ERR= present
! depending on OPEN[OPEN$K_ERR].  Pass UNWIND action code.
!-

    ENABLE
        FOR$$ERR_OPECLO (L_UNWIND_ACTION, OPEN);

!+
! Set UNWIND cleanup to be a no-operation since LUB/ISB/RAB
! has not been pushed yet.
!-
    L_UNWIND_ACTION = FOR$K_UNWINDNOP;
!+
! Copy user keyword arglist into array OPEN
! in cannonical order, so that args may be processed in order
! If ASCIZ name string, setup NAM_DSC as its descriptor
! If ASCIZ default name string, setup DEF_DSC as its descriptor
! SIGNAL_STOP FOR$_INVARGFOR (48='INVALID ARGUMENT TO FORTRAN I/O SYSTEM'),
! after scanning all parameters and setting up ERR= in OPEN array.
```

```
2 !-
2       FOR$$OPECLO_ARG (KEYWD, ACTUALCOUNT (), OPEN, OPEN$K_KEY_MAX, NAM_DSC, DEF_DSC, 1);
2 !+
2 ! Allocate LUB/ISB/RAB if not already allocated for this
2 ! logical unit. Push down if an I/O statement already in progress
2 ! on another unit. Store new current LUB address in OTS
2 ! GLOBAL OWN OTS$A_CUR_LUB. SIGNAL_STOP FOR$_RECIO_OPE
2 ! (40='RECURSIVE I/O OPERATION'). If an I/O statement already
2 ! in progress for this logical unit. SIGNAL_STOP FOR$_INVLOGUNI
2 ! (32='INVALID LOGICAL UNIT NUMBER') if logical unit
2 ! number outside of the allowed range of 0:99 for explicit OPEN.
2 ! Finally change UNWIND cleanup action to be to pop current LUB/ISB/RAB
2 ! since it has now been successfully pushed.
2 ! On return, CCB points to the current control block.
2 !-
2       FOR$$CB_PUSH (.OPEN [OPEN$K_UNIT], LUB$K_LUN_MIN);
2       L_UNWIND_ACTION = FOR$K_UNWINDPOP;
2 !+
2 ! If the unit is currently open, call special routine which
2 ! implements open on a connected unit.
2 !-
2
3       IF (.CCB [LUB$V_OPENED] OR .CCB [LUB$V_DEALLOC])
2       THEN
2
2           IF OPEN_ON_CONNECTED (OPEN, L_UNWIND_ACTION)
2           THEN
3               BEGIN
3 !+
3 ! No more OPEN processing needed, set IOSTAT and exit.
3 !-
3
4               IF (.OPEN [OPEN$K_IOSTAT] NEQ 0)
3               THEN
4                   BEGIN
4
5                   IF (.OPEN [OPEN$K_IOSTAT_L])
4                   THEN
4                       .OPEN [OPEN$K_IOSTAT] = 0
4                   ELSE
5                       BEGIN
5
5                       LOCAL
5                           IOSTAT : REF BLOCK [, BYTE];
5
5                       IOSTAT = .OPEN [OPEN$K_IOSTAT];
5                       IOSTAT [0, 0, 16, 0] = 0;   ! Store one word
4                       END;
4
3                   END;
3
3               RETURN 1;                           ! Exit OPEN successfully
2               END;
2
2 !+
2 ! If DEFINE FILE, CALL FDBSET, or CALL ASSIGN have already been
2 ! done for this logical unit, SIGNAL_STOP FOR$DUPFILSPE
```

```
2 ! (21='DUPLICATE FILE SPECIFICATION').
2 !-
2
2     IF ((.CCB [LUB$A_FAB] NEQA 0) OR (.CCB [LUB$V_DIRECT])) THEN FOR$$SIGNAL_STO (FOR$K_DUPFILSPE);
2
2 !+
2 ! Set unwind condition to RET so if an error occurs the file will
2 ! be closed and the LUB returned (thus freeing up the LUN).
2 !-
2     L_UNWIND_ACTION = FOR$K_UNWINDRET;
2 !+
2 ! Perform the OPEN - call common procedure with a pointer
2 ! to the OPEN parameter VECTOR of longword values.
2 !-
2     FOR$$OPEN_PROC (OPEN);
2 !+
2 ! Pop back previous LUB or indicate that no I/O statement
2 ! is currently active (OTS$$A_CUR_LUB = 0).
2 !-
2     FOR$$CB_POP ();
2 !+
2 !  Store success IOSTAT.  If there was an error, the handler would
2 !  do the store.
2 !-
2
3     IF (.OPEN [OPEN$K_IOSTAT] NEQ 0)
2     THEN
2
3         IF (.OPEN [OPEN$K_IOSTAT_L])
2         THEN
2             .OPEN [OPEN$K_IOSTAT] = 0
2         ELSE
3             BEGIN
3
3             LOCAL
3                 IOSTAT : REF BLOCK [, BYTE];
3
3             IOSTAT = .OPEN [OPEN$K_IOSTAT];
3             IOSTAT [0, 0, 16, 0] = 0;                 ! Store one word
2             END;
2
2 !+
2 ! Return success
2 !-
2     RETURN 1;
1     END;                                              ! End of FOR$OPEN routine

                                                          .TITLE  FOR$OPEN FORTRAN OPEN
                                                          .IDENT  \1-065\

                                                          .EXTRN  FOR$$ERR_OPECLO
                                                          .EXTRN  FOR$$OPEN_PROC, FOR$$SIGNAL_STO
                                                          .EXTRN  FOR$$SIG_NO_LUB
                                                          .EXTRN  FOR$$CB_PUSH, FOR$$CB_POP
                                                          .EXTRN  FOR$$OPEN_KEYWD
                                                          .EXTRN  FOR$$SIG_FATINT
```

```
                                                     .EXTRN  FOR$$CLOSE_FILE

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

                                 0804 00000          .ENTRY  FOR$OPEN, Save R2,R11                   : 0168
            5E         88    AE   9E 00002          MOVAB   -120(SP), SP
                             7E   7C 00006          CLRQ    OPEN                                     : 0241
                       08    AE   7C 00008          CLRQ    OPEN
                       10    AE   7C 0000B          CLRQ    OPEN
                       18    AE   7C 0000E          CLRQ    OPEN
                       20    AE   7C 00011          CLRQ    OPEN
                       28    AE   7C 00014          CLRQ    OPEN
                       30    AE   7C 00017          CLRQ    OPEN
                       38    AE   7C 0001A          CLRQ    OPEN
                       40    AE   7C 0001D          CLRQ    OPEN
                       48    AE   7C 00020          CLRQ    OPEN
                       50    AE   7C 00023          CLRQ    OPEN
                       58    AE   7C 00026          CLRQ    OPEN
                       60    AE   7C 00029          CLRQ    OPEN
                       68    AE   7C 0002C          CLRQ    OPEN
            6D       0081    CF   DE 0002F          MOVAL   8$, (FP)
        6C  AE               01   D0 00034          MOVL    #1, L_UNWIND_ACTION                      : 0274
                             01   DD 00038          PUSHL   #1                                       : 0283
                       74    AE   9F 0003A          PUSHAB  DEF_DSC
                       F8    AD   9F 0003D          PUSHAB  NAM_DSC
                             1A   DD 00040          PUSHL   #26
                       10    AE   9F 00042          PUSHAB  OPEN
            7E               6C   9A 00045          MOVZBL  (AP), -(SP)
                       04    AC   9F 00048          PUSHAB  KEYWD
     0000V  CF               07   FB 0004B          CALLS   #7, FOR$$OPECLO_ARG
                             50   D4 00050          CLRL    R0                                       : 0297
            52         04    AE   D0 00052          MOVL    OPEN+4, R2
                00000000G    00   16 00056          JSB     FOR$$CB_PUSH
                       6C    AE   D4 0005C          CLRL    L_UNWIND_ACTION                          : 0298
            05         FC    AB   E8 0005F          BLBS    -4(CCB), 1$                              : 0304
   0E       FF  AB               04   E1 00063          BBC     #4, -1(CCB), 2$
                       6C    AE   9F 00068 1$:      PUSHAB  L_UNWIND_ACTION                          : 0307
                       04    AE   9F 0006B          PUSHAB  OPEN
     0000V  CF               02   FB 0006E          CALLS   #2, OPEN_ON_CONNECTED
            26               50   E8 00073          BLBS    R0, 5$
                       E8    AB   D5 00076 2$:      TSTL    -24(CCB)                                 : 0342
                             05   12 00079          BNEQ    3$
   09       FC  AB               04   E1 0007B          BBC     #4, -4(CCB), 4$
                             15   DD 00080 3$:      PUSHL   #21
 00000000G  00               01   FB 00082          CALLS   #1, FOR$$SIGNAL_STO
        6C  AE               02   D0 00089 4$:      MOVL    #2, L_UNWIND_ACTION                      : 0348
                             5E   DD 0008D          PUSHL   SP                                       : 0353
 00000000G  00               01   FB 0008F          CALLS   #1, FOR$$OPEN_PROC
                00000000G    00   16 00096          JSB     FOR$$CB_POP                              : 0358
                       58    AE   D5 0009C 5$:      TSTL    OPEN+88                                  : 0364
                             0F   13 0009F          BEQL    7$
            05         64    AE   E9 000A1          BLBC    OPEN+100, 6$                             : 0367
                       58    BE   D4 000A5          CLRL    @OPEN+88                                 : 0369
                             06   11 000A8          BRB     7$
            50         58    AE   D0 000AA 6$:      MOVL    OPEN+88, IOSTAT                          : 0376
                             60   B4 000AE          CLRW    (IOSTAT)                                 : 0377
            50               01   D0 000B0 7$:      MOVL    #1, R0                                   : 0383
```

```
                                          04 000B3         RET                                          ; 0384
                                        0000 000B4 8$:     .WORD   Save nothing                         ; 0241
                      50       08    AC   D0 000B6         MOVL    8(AP), R0                            ;
                      50       04    A0   D0 000BA         MOVL    4(R0), R0                            ;
                               80    A0   9F 000BE         PUSHAB  OPEN                                 ;
                               EC    A0   9F 000C1         PUSHAB  L_UNWIND_ACTION                      ;
                                     02   DD 000C4         PUSHL   #2                                   ;
                                     5E   DD 000C6         PUSHL   SP                                   ;
                      7E       04    AC   7D 000C8         MOVQ    4(AP), -(SP)                         ;
           00000000G  00             03   FB 000CC         CALLS   #3, FOR$$ERR_OPECLO                  ;
                                          04 000D3         RET                                          ;

; Routine Size:  212 bytes,    Routine Base:  _FOR$CODE + 0000

;  322        0385  1
```

```
GLOBAL ROUTINE FOR$$OPECLO_ARG (                        ! FORTRAN copy OPEN/CLOSE args
        KEYWD_ADR,                                      ! Adr. of first keyword arg
        ACTUAL_COUNT,                                   ! No. of actual parameters in arg list
        OPEN_ADR,                                       ! Adr. of array to store keyword values
        KEY_MAX,                                        ! Max. value of keyword parameter
        NAM_DSC_ADR,                                    ! Adr. of descriptor for name string
        DEF_DSC_ADR,                                    ! Adr. of descriptor for default name string
        OPEN_FLAG,                                      ! True if OPEN (not CLOSE)
        VAR_LENGTHS                                     ! lengths in bits of keyword variables
    ) : NOVALUE =

!++
! ABSTRACT:
!
!       Routine to copy keyword OPEN/CLOSE parameters
!       into an array for sequential processing in cannonical order.
!       Note: LUB cannot be located until all OPEN arguments are scanned and UNIT=n found.
!
! FORMAL PARAMETERS:
!
!       KEYWD_ADR.rlu.ra        Address of first keyword
!                               in user arg list
!       ACTUAL_COUNT.rlu.v      Count of no. of users args
!       OPEN_ADR.wlu.ra         Adr. of array to write keyword values
!       KEY_MAX.rlu.v   Max. OPEN/CLOSE keyword value
!       NAM_DSC_ADR     Adr. of a descriptor if ASCIZ name string given by user
!       DEF_DSC_ADR     Adr. of a descriptor if ASCIZ default name string given by user
!                       Descriptors must be allocated by caller
!                       not called procedure.
!       OPEN_FLAG       = 1 if this call is from OPEN, 0 from CLOSE.
!                       Only allocate a LUN if from OPEN.
!       VAR_LENGTHS     A byte vector into which are inserted the lengths
!                       in bits of the keyword variables.  This is used
!                       by FOR$INQUIRE only.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETITION STATUS:
!
!       NONE
!
! SIDE EFFECTS:
!
!       SIGNAL_STOPs FOR$_INVARGFOR (48='INVALID ARGUMENT TO FORTRAN I/O SYSTEM')
!       if keyword parameter is out of range, but only after all parameters
!       are scanned so that ERR= parameter, if present, has been setup in array OPEN_ADR.
!       Uses FOR$$SIG_NO_LUB to signal, since no LUB setup yet
!       so logical unit number must be passed explicitly on errors.
!--

    BEGIN
```

```
2
2        MAP
2            KEYWD_ADR : REF BLOCKVECTOR [100, 1],      ! Vector of blocks, each block
2                                                       ! is one longword.
2            OPEN_ADR : REF VECTOR [OPEN$K_KEY_MAX + 1, LONG],     ! Vector to receive cannonical ordering
2                                                       ! of users parameter values.
2            NAM_DSC_ADR : REF DSC$DESCRIPTOR,          ! string descriptor to use in case ASCIZ file name
2            DEF_DSC_ADR : REF DSC$DESCRIPTOR,          ! string descriptor to use in case ASCIZ default file name
2            VAR_LENGTHS : REF VECTOR [INQ$K_KEY_MAX + 1, BYTE];     ! Variable lengths
2
2        LOCAL
2            V_ARG_KEY_ERR,                             ! error flag, 1 if ARG or KEY out of range
2            V_KEY_VAL_ERR,                             ! error flag, 1 if keyword incorrect
2            UNIT_ADDR,                                 ! Address of UNIT variable
2            UNIT_TYPE;                                 ! Type of varaible: w or L
2
2    !+
2    ! Clear OPEN or CLOSE parameter array and clear flag
2    !-
2        FILL_VAL (0, .KEY_MAX + 1, .OPEN_ADR);
2        V_ARG_KEY_ERR = 0;
2        V_KEY_VAL_ERR = 0;
2        UNIT_TYPE = 0;
2        UNIT_ADDR = 0;
2    !+
2    ! Scan actual keyword parameter list (KEYWD_ADR) and copy (sign extend)
2    ! associated information to formal array OPEN_ADR of longwords ordered
2    ! by parameter dependencies, i. e., sort by KEY.
2    !-
2
2        INCR I FROM 0 TO .ACTUAL_COUNT - 1 DO
2    !+
2    ! Set longword value to sign extension of each type of OPEN/CLOSE
2    ! parameter present to: Bits 31:16 of this actual, next
2    ! actual, or location specified by next actual depending
2    ! on the type of OPEN argument (OPEN$B_ARG_TYPE).
2    ! If ARGTYPE or KEY code is not one of defined values, set error flag and keep scanning
2    ! to see if ERR= is present so error handler will handle properly.
2    ! error FOR$_INVARGFOR (48='INVALID ARGUMENT TO FORTRAN I/O SYSTEM')
2    !-
3            BEGIN
3
3            LOCAL
3                K,                                     ! temp value of KEY
3                V;                                     ! temp value of value to be stored
3
3            K = .KEYWD_ADR [.I, OPEN$B_KEY];
3            V =
4            BEGIN
4
4            CASE .KEYWD_ADR [.I, OPEN$B_ARG_TYPE] FROM 0 TO OPEN$K_ARG_MAX OF
4                SET
4
4                [OPEN$K_ARG_NULL] :
4    !+
4    ! keyword with no value - make value be 1
4    ! to distinguish from not present.
```

```
4 !-
4                       1;
4
4               [OPEN$K_ARG_LIT, OPEN$K_ARG_W_V] :
4 !+
4 ! literal or word-by-value - bits <31:16> is value
4 ! sign extend to full machine value
4 !-
4                       .KEYWD_ADR [.I, OPEN$W_INFO];
4
4               [OPEN$K_ARG_W_R] :
4 !+
4 ! Word by reference - use adr. in next longword
4 ! sign extend word to longword
4 !-
5                       BEGIN
5
6                       IF (.K EQLU OPEN$K_UNIT)
5                       THEN
5 !+
5 ! Remember UNIT's address and type in case we must provide it
5 !-
5
6                           IF (.UNIT_TYPE NEQ 0)
5                           THEN
5                               V_ARG_KEY_ERR = 1
5                           ELSE
6                               BEGIN
6 !+
6 ! This is the first time through here
6 !-
6                               UNIT_TYPE = DSC$K_DTYPE_W;
6                               UNIT_ADDR = .KEYWD_ADR [.I + 1, OPEN$A_VALUE];
5                               END;
5
6                       IF ((.K EQLU OPEN$K_ASSOCIAT) OR (.K EQLU OPEN$K_IOSTAT))
5                       THEN
5 !+
5 ! For the associated variable or IOSTAT we want the address of the value, not the
5 ! value itself.
5 !-
5                           .KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE]      !
5                       ELSE
5
6                           IF (.K GTR OPEN$K_KEY_MAX)
5                           THEN
6                               BEGIN
6                               VAR_LENGTHS [.K] = 16;   ! Signify word
6                               .KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE]
6                               END
5                           ELSE
5                               .(.KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE])<0, %BPVAL/2, 1>
5
4                       END;
4
4               [OPEN$K_ARG_L_R] :
4 !+
4 ! Longword by-reference-next parameter slot contains adr. of value
4 !-
5               BEGIN
5
6               IF (.K EQLU OPEN$K_UNIT)
5               THEN
5 !+
5 ! Remember the address and type of the variable which holds the UNIT
5 ! in case we must compute the LUN value.
5 !-
5
6                   IF (.UNIT_TYPE NEQ 0)
5                   THEN
5                       V_ARG_KEY_ERR = 1
5                   ELSE
6                       BEGIN
6 !+
6 ! This is the first time through here.
6 !-
6                       UNIT_TYPE = DSC$K_DTYPE_L;
6                       UNIT_ADDR = .KEYWD_ADR [.I + 1, OPEN$A_VALUE];
5                       END;
5
6               IF ((.K EQLU OPEN$K_ASSOCIAT) OR (.K EQLU OPEN$K_IOSTAT))
5               THEN
5 !+
5 ! For the associated variable or IOSTAT we want the address of the variable, not
5 ! its value.  Also, we must mark that it occupies a longword.
5 !-
6                   BEGIN
6
7                   IF (.K EQLU OPEN$K_ASSOCIAT)
6                   THEN
6                       OPEN_ADR [OPEN$K_ASSOC_L] = 1
6                   ELSE
6                       OPEN_ADR [OPEN$K_IOSTAT_L] = 1;
6
6                   .KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE]
6                   END
5               ELSE
5
6                   IF (.K GTR OPEN$K_KEY_MAX)
5                   THEN
6                       BEGIN
6                       VAR_LENGTHS [.K] = 32;  ! Signify longword
6                       .KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE] ! Address for INQUIRE
6                       END
5                   ELSE
5                       ..KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE]
5
4               END;
4
4           [OPEN$K_ARG_L_V, OPEN$K_ARG_ZI] :
4 !+
4 ! Longword by value or procedure adr.
4 !-
4               .KEYWD_ADR [(I = .I + 1), OPEN$G_VALUE];
```

```

            [OPEN$K_ARG_T_DS] :
!+
! Address of string descriptor.
!-

                IF .K EQLU OPEN$K_NAME OR .K EQLU OPEN$K_DEFAULTF
                THEN
                    .KEYWD_ADR [(I = .I + 1), OPEN$G_VALUE]
                ELSE
                    BEGIN

                    LOCAL
                        V;                          ! Returned value

                    V = FOR$$OPEN_KEYWD (.K, .KEYWD_ADR [.I + 1, OPEN$G_VALUE]);
                    I = .I + 1;

                    CASE .V FROM -1 TO 0 OF
                        SET

                        [-1] :                      ! Invalid keyword for this type
                            BEGIN
                            V_ARG_KEY_ERR = 1;
                            0
                            END;

                        [0] :                       ! Keyword value error
                            BEGIN
                            V_KEY_VAL_ERR = 1;
                            0
                            END;

                        [OUTRANGE] :                ! Ok
                            .V;
                        TES

                    END;

            [OPEN$K_ARG_TZ_R] :
!+
! Address of array of ASCIZ characters.
! Next parameter slot contains address of first byte of string
! If this is FILE or DEFAULTFILE, store length and address of string in
! its respective descriptor.
! Else SIGNAL_STOP FOR$_INVARGFOR (48='INVALID ARGUMENT TO FORTRAN I/O SYSTEM')
!-

                IF (.K EQLU OPEN$K_NAME)
                THEN
                    BEGIN

                    LOCAL
                        P;                          ! char. pointer to null char or 0

                    NAM_DSC_ADR [DSC$A_POINTER] = .KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE];
                    P = CH$FIND_CH (OPEN$K_STR_MAX, .NAM_DSC_ADR [DSC$A_POINTER], 0);
```

```
6                       NAM_DSC_ADR [DSC$W_LENGTH] = (IF .P NEQ 0 THEN CH$DIFF (.P, .NAM_DSC_ADR [DSC$A_POINTER]
5                       ELSE OPEN$K_STR_MAX);
5                       .NAM_DSC_ADR                    ! value of the CASE-expr is adr. of descr.
5                       END
5               ELSE IF (.K EQLU OPEN$K_DEFAULTF)
4               THEN
5                       BEGIN
5
5                       LOCAL
5                           P;                          ! char. pointer to null char or 0
5
5                       DEF_DSC_ADR [DSC$A_POINTER] = .KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE];
5                       P = CH$FIND_CH (OPEN$K_STR_MAX, .DEF_DSC_ADR [DSC$A_POINTER], 0);
6                       DEF_DSC_ADR [DSC$W_LENGTH] = (IF .P NEQ 0 THEN CH$DIFF (.P, .DEF_DSC_ADR [DSC$A_POINTER]
5                       ELSE OPEN$K_STR_MAX);
5                       .DEF_DSC_ADR                    ! value of the CASE-expr is adr. of descr.
5                       END
4               ELSE
4   !+
4   ! ASCIZ string not file name or default file name, just skip next
4   ! longword and flag error
4   !-
5                       BEGIN
5                       I = .I + 1;
5                       V_ARG_KEY_ERR = 1;
5                       0                               ! value of the CASE-expr is 0 if error
4                       END;
4
4           [OPEN$K_ARG_INLN] :
4   !+
4   ! Sublist in-line with argument list
4   !-
5               BEGIN
5
5               LOCAL
5                   ADDR,
5                   COUNT;
5
5               COUNT = .KEYWD_ADR [.I, OPEN$W_INFO];
5               ADDR = KEYWD_ADR [.I, OPEN$B_KEY];
5               I = .I + .COUNT;
5               .ADDR
4               END;
4
4           [OPEN$K_ARG_B_R] :
4   !+
4   ! Byte variable by reference
4   ! Used only by FOR$INQUIRE
4   !-
5               BEGIN
5
6               IF (.K GTR OPEN$K_KEY_MAX)
5               THEN
6                   BEGIN
6                   VAR_LENGTHS [.K] = 8;       ! Signify byte
6                   .KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE]
6                   END
```

```
;    666          0728 5                  ELSE
;    667          0729 5                      ..KEYWD_ADR [(I = .I + 1), OPEN$A_VALUE]
;    668          0730 5
;    669          0731 4                  END;
;    670          0732 4
;    671          0733 4              [INRANGE, OUTRANGE] :
;    672          0734 4  !+
;    673          0735 4  ! If KEY is out of range, set error flag (V_ARG_KEY_ERR) and
;    674          0736 4  ! keep scanning to see if ERR= is present or not.
;    675          0737 4  !-
;    676          0738 5                  BEGIN
;    677          0739 5                  V_ARG_KEY_ERR = 1;
;    678          0740 5                  0                                     ! Store 0
;    679          0741 4                  END;
;    680          0742 4              TES
;    681          0743 4
;    682          0744 3          END;                                          ! End of CASE on ARG_TYPE.
;    683          0745 3  !+
;    684          0746 3  ! If KEY value is in range, store in cannonical array OPEN_ADR,
;    685          0747 3  ! else set error flag and keep scanning to see if ERR= is present
;    686          0748 3  ! so error handler will handle properly when signaled.
;    687          0749 3  ! Note: I advanced correctly (by 1 or 2) depending on ARGTYPE
;    688          0750 3  ! even though KEY is not one of the defined ones.
;    689          0751 3  !-
;    690          0752 3
;    691          0753 3          IF ((.K LEQU .KEY_MAX) OR (.K EQLU OPEN$K_IOSTAT)) THEN OPEN_ADR [.K] = .V ELSE V_ARG_KEY_ERR = 1;
;    692          0754 3
;    693          0755 2          END;                                          ! End of loop
;    694          0756 2
;    695          0757 2  !+
;    696          0758 2  ! Check for any errors during scan.
;    697          0759 2  ! If yes, SIGNAL_STOP FOR$_INVARGFOR (48='INVALID ARGUMENT TO FORTRAN I/O SYSTEM')
;    698          0760 2  !-
;    699          0761 2
;    700          0762 2      IF .V_ARG_KEY_ERR THEN FOR$$SIG_NO_LUB (FOR$K_INVARGFOR, .OPEN_ADR [OPEN$K_UNIT]);
;    701          0763 2
;    702          0764 2      IF .V_KEY_VAL_ERR THEN FOR$$SIG_NO_LUB (FOR$K_KEYVALERR, .OPEN_ADR [OPEN$K_UNIT]);
;    703          0765 2
;    704          0766 2
;    705          0767 2      RETURN;                                           ! Return from FOR$$OPECLO_ARG routine
;    706          0768 1      END;                                              ! End of FOR$$OPECLO_ARG routine



                                            0FFC 00000        .ENTRY  FOR$$OPECLO_ARG, Save R2,R3,R4,R5,R6,R7,R8,-; 0386
                                                                      R9,R10,R11                                   ;
                                  5E     04  C2 00002         SUBL2   #4, SP                                      ;
                                  5B  10 AC  D0 00005         MOVL    KEY_MAX, R11                                ; 0462
                    50            5B     02  78 00009         ASHL    #2, R11, R0                                 ;
                                  50     04  C0 0000D         ADDL2   #4, R0                                      ;
                                  57  0C AC  D0 00010         MOVL    OPEN_ADR, R7                                ;
          50        00            6E     00  2C 00014         MOVC5   #0, (SP), #0, R0, (R7)                      ;
                                         67     00019                                                            ;
                                         6E  D4 0001A         CLRL    V_KEY_VAL_ERR                               ; 0464
                                         58  7C 0001C         CLRQ    UNIT_TYPE                                   ; 0465
                                  5A  D4 0001E          CLRL    UNIT_ADDR                        ; 0466
                    55      04    AC  D0 00020          MOVL    KEYWD_ADR, R5                    ; 0489
                    52            01  CE 00024          MNEGL   #1, I
                                0175  31 00027          BRW     42$
                    50          6542  DE 0002A  1$:     MOVAL   (R5)[I], R0
                    53            60  9A 0002E          MOVZBL  (R0), K
            0A      00      01    A0  8F 00031          CASEB   1(R0), #0, #10                   ; 0493
 0025     001E    001E          0019     00036  2$:     .WORD   3$-2$,-
 00AE     00DA    005C          0145     0003E                  4$-2$,-
          013B    012F          0145     00046                  4$-2$,-
                                                                6$-2$,-
                                                                37$-2$,-
                                                                10$-2$,-
                                                                25$-2$,-
                                                                20$-2$,-
                                                                37$-2$,-
                                                                34$-2$,-
                                                                36$-2$
                                010F  31 0004C          BRW     32$                              ; 0739
                    54            01  D0 0004F  3$:     MOVL    #1, V                            ; 0493
                                  04  11 00052          BRB     5$
                    54      02    A0  32 00054  4$:     CVTWL   2(R0), V                         ; 0508
                                0131  31 00058  5$:     BRW     39$
                    01            53  D1 0005B  6$:     CMPL    K, #1                            ; 0517
                                  11  12 0005E          BNEQ    8$
                                  58  D5 00060          TSTL    UNIT_TYPE                        ; 0523
                                  05  13 00062          BEQL    7$
                    59            01  D0 00064          MOVL    #1, V_ARG_KEY_ERR                ; 0525
                                  08  11 00067          BRB     8$
                    58            07  D0 00069  7$:     MOVL    #7, UNIT_TYPE                    ; 0531
                    5A      04  A542  D0 0006C          MOVL    4(R5)[I], UNIT_ADDR              ; 0532
                    11            53  D1 00071  8$:     CMPL    K, #17                           ; 0535
                                  76  13 00074          BEQL    21$
                    16            53  D1 00076          CMPL    K, #22
                                  71  13 00079          BEQL    21$
                    1A            53  D1 0007B          CMPL    K, #26                           ; 0544
                                  07  15 0007E          BLEQ    9$
               20 BC43            10  90 00080          MOVB    #16, @VAR_LENGTHS[K]             ; 0547
                                  4A  11 00085          BRB     18$                              ; 0548
                                  52  D6 00087  9$:     INCL    I                                ; 0551
                    50          6542  D0 00089          MOVL    (R5)[I], R0
                    50            60  32 0008D          CVTWL   (R0), R0
                                  77  11 00090          BRB     23$                              ; 0544
                    01            53  D1 00092  10$:    CMPL    K, #1                            ; 0561
                                  11  12 00095          BNEQ    12$
                                  58  D5 00097          TSTL    UNIT_TYPE                        ; 0568
                                  05  13 00099          BEQL    11$
                    59            01  D0 0009B          MOVL    #1, V_ARG_KEY_ERR                ; 0570
                                  08  11 0009E          BRB     12$
                    58            08  D0 000A0  11$:    MOVL    #8, UNIT_TYPE                    ; 0576
                    5A      04  A542  D0 000A3          MOVL    4(R5)[I], UNIT_ADDR              ; 0577
                                  50  D4 000A8  12$:    CLRL    R0                               ; 0580
                    11            53  D1 000AA          CMPL    K, #17
                                  04  12 000AD          BNEQ    13$
                                  50  D6 000AF          INCL    R0
                                  05  11 000B1          BRB     14$
                    16            53  D1 000B3  13$:    CMPL    K, #22
```

```
                                       0F  12 000B6            BNEQ    17$                                    ;
                               05      50  E9 000B8  14$:      BLBC    R0, 15$                                ; 0588
                               67      01  D0 000BB            MOVL    #1, (R7)                               ; 0590
                                       04  11 000BE            BRB     16$                                    ;
                          64   A7      01  D0 000C0  15$:      MOVL    #1, 100(R7)                            ; 0592
                                     00B4  31 000C4  16$:      BRW     37$                                    ; 0594
                               1A      53  D1 000C7  17$:      CMPL    K, #26                                 ; 0598
                                       0D  15 000CA            BLEQ    19$                                    ;
                         20 BC43       20  90 000CC            MOVB    #32, @VAR_LENGTHS[K]                   ; 0601
                                       52  D6 000D1  18$:      INCL    I                                      ; 0602
                               50    6542  D0 000D3            MOVL    (R5)[I], R0                            ;
                                       30  11 000D7            BRB     23$                                    ;
                                       52  D6 000D9  19$:      INCL    I                                      ; 0605
                               50    6542  D0 000DB            MOVL    (R5)[I], R0                            ;
                               50      60  D0 000DF            MOVL    (R0), R0                               ;
                                       25  11 000E2            BRB     23$                                    ; 0598
                               0E      53  D1 000E4  20$:      CMPL    K, #14                                 ; 0620
                                       DB  13 000E7            BEQL    16$                                    ;
                               1A      53  D1 000E9            CMPL    K, #26                                 ;
                                       D6  13 000EC  21$:      BEQL    16$                                    ;
                                  04 A542  DD 000EE            PUSHL   4(R5)[I]                               ; 0629
                                       53  DD 000F2            PUSHL   K                                      ;
                      00000000G   00   02  FB 000F4            CALLS   #2, FOR$$OPEN_KEYWD                    ;
                                       52  D6 000FB            INCL    I                                      ; 0630
                    01 FFFFFFFF   8F   50  CF 000FD            CASEL   V, #-1, #1                             ; 0632
                             0006    0059     00105  22$:      .WORD   32$-22$,-                              ;
                                                                       24$-22$                                ;
                                       61  11 00109  23$:      BRB     35$                                    ; 0648
                               6E      01  D0 0010B  24$:      MOVL    #1, V_KEY_VAL_ERR                      ; 0643
                                       51  11 0010E            BRB     33$                                    ; 0642
                               0E      53  D1 00110  25$:      CMPL    K, #14                                 ; 0662
                                       16  12 00113            BNEQ    26$                                    ;
                               56   14 AC  D0 00115            MOVL    NAM_DSC_ADR, R6                        ; 0669
                                       52  D6 00119            INCL    I                                      ;
                            04 A6    6542  D0 0011B            MOVL    (R5)[I], 4(R6)                         ;
                 04   B6   0064 8F     00  3A 00120            LOCC    #0, #100, @4(R6)                       ; 0670
                                       1B  13 00127            BEQL    27$                                    ;
                                       1B  11 00129            BRB     28$                                    ; 0671
                               1A      53  D1 0012B  26$:      CMPL    K, #26                                 ; 0675
                                       2C  12 0012E            BNEQ    31$                                    ;
                               56   18 AC  D0 00130            MOVL    DEF_DSC_ADR, R6                        ; 0682
                                       52  D6 00134            INCL    I                                      ;
                            04 A6    6542  D0 00136            MOVL    (R5)[I], 4(R6)                         ;
                 04   B6   0064 8F     00  3A 0013B            LOCC    #0, #100, @4(R6)                       ; 0683
                                       02  12 00142            BNEQ    28$                                    ;
                                       51  D4 00144  27$:      CLRL    R1                                     ;
                                       51  D5 00146  28$:      TSTL    P                                      ; 0684
                                       06  13 00148            BEQL    29$                                    ;
                               51   04 A6  C2 0014A            SUBL2   4(R6), R1                              ;
                                       04  11 0014E            BRB     30$                                    ;
                               51   64 8F  9A 00150  29$:      MOVZBL  #100, R1                               ;
                               66      51  B0 00154  30$:      MOVW    R1, (R6)                               ;
                               54      56  D0 00157            MOVL    R6, V                                  ; 0686
                                       30  11 0015A            BRB     39$                                    ;
                                       52  D6 0015C  31$:      INCL    I                                      ; 0694
                               59      01  D0 0015E  32$:      MOVL    #1, V_ARG_KEY_ERR                      ; 0695
                                       54  D4 00161  33$:      CLRL    V                                      ; 0693
```

```
                                    27  11 00163           BRB     39$                                       ; 0662
                 51        02       A0  32 00165 34$:      CVTWL   2(R0), COUNT                              ; 0709
                 52                 51  C0 00169           ADDL2   COUNT, I                                  ; 0711
                 54                 50  D0 0016C 35$:      MOVL    ADDR, V                                   ; 0712
                                    1B  11 0016F           BRB     39$                                       ;
                 1A                 53  D1 00171 36$:      CMPL    K, #26                                    ; 0722
                                    0D  15 00174           BLEQ    38$                                       ;
            20 BC43                 08  90 00176           MOVB    #8, @VAR_LENGTHS[K]                       ; 0725
                                    52  D6 0017B 37$:      INCL    I                                         ; 0726
                 54               6542  D0 0017D           MOVL    (R5)[I], V                                ;
                                    09  11 00181           BRB     39$                                       ;
                                    52  D6 00183 38$:      INCL    I                                         ; 0729
                 50               6542  D0 00185           MOVL    (R5)[I], R0                               ;
                 54                 60  D0 00189           MOVL    (R0), V                                   ;
                 5B                 53  D1 0018C 39$:      CMPL    K, R11                                    ; 0753
                                    05  1B 0018F           BLEQU   40$                                       ;
                 16                 53  D1 00191           CMPL    K, #22                                    ;
                                    06  12 00194           BNEQ    41$                                       ;
               6743                 54  D0 00196 40$:      MOVL    V, (R7)[K]                                ;
                                    03  11 0019A           BRB     42$                                       ;
                 59                 01  D0 0019C 41$:      MOVL    #1, V_ARG_KEY_ERR                         ;
         02      52        08       AC  F2 0019F 42$:      AOBLSS  ACTUAL_COUNT, I, 43$                      ; 0473
                                    03  11 001A4           BRB     44$                                       ;
                                  FE81  31 001A6 43$:      BRW     1$                                        ;
                 0C                 59  E9 001A9 44$:      BLBC    V_ARG_KEY_ERR, 45$                        ; 0762
                           04       A7  DD 001AC           PUSHL   4(R7)                                     ;
                                    30  DD 001AF           PUSHL   #48                                       ;
          00000000G  00             02  FB 001B1           CALLS   #2, FOR$$SIG_NO_LUB                       ;
                 0C                 6E  E9 001B8 45$:      BLBC    V_KEY_VAL_ERR, 46$                        ; 0764
                           04       A7  DD 001BB           PUSHL   4(R7)                                     ;
                                    2D  DD 001BE           PUSHL   #45                                       ;
          00000000G  00             02  FB 001C0           CALLS   #2, FOR$$SIG_NO_LUB                       ;
                                        04 001C7 46$:      RET                                               ; 0768

; Routine Size:  456 bytes,    Routine Base:  _FOR$CODE + 00D4

;  707          0769  1
```

```
1 ROUTINE OPEN_ON_CONNECTED (                          ! Open on a connected unit
1           OPEN,                                      ! Keyword vector
1           L_UNWIND_ACTION                            ! Unwind action
1     ) : CALL_CCB =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         This routine implements the FORTRAN-77 concept of open on
1 !         a connected unit.
1 !
1 !         If an OPEN is done for a unit which is already open, one of two
1 !         things happen:
1 !             1.  If the filename specification in the OPEN is the same as
1 !                 the same as the file currently open, or if the filename
1 !                 is omitted but the unit is already open, then the value
1 !                 of BLANK= is set according to the keyword list.
1 !             2.  If the filename specification in the OPEN is not the same
1 !                 as the file currently open, the old file is closed and
1 !                 the new one is opened.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         OPEN.rl.ra              Sorted keyword list from OPEN
1 !         L_UNWIND_ACTION.ml.r    Unwind action in case of an error
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         CCB                     Global I/O database register
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         LUB$V_NULLBLNK
1 !
1 ! ROUTINE VALUE:
1 !
1 !         True (1) if no further OPEN processing is needed (case 1)
1 !         False (0) otherwise (case 2)
1 !
1 ! SIDE EFFECTS:
1 !
1 !         Possibly closes the currently open file
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         CCB : REF $FOR$CCB_DECL;
2
2     MAP
2         OPEN : REF VECTOR [OPEN$K_KEY_MAX + 1];
2
2     LOCAL
2         FAB : BLOCK [FAB$C_BLN, BYTE],               ! FAB block
2         NAM : BLOCK [NAM$C_BLN, BYTE],               ! NAM block
2         RES_NAME : VECTOR [NAM$C_MAXRSS, BYTE],      ! Resultant name string
2         RES_LEN,                                     ! Resultant string length
```

```
2               DEF_NAME : VECTOR [10, BYTE],           ! Default name string
2               NAM_DSC : REF DSC$DESCRIPTOR,           ! FILE/DEFAULTFILE descriptor
2               UNIT,                                   ! Logical unit number
2               RMS_STATUS;                             ! RMS condition status
2
2       !+
2       ! Set up FAB and NAM blocks
2       !-
2           CH$FILL (0, FAB$C_BLN, FAB);
2           CH$FILL (0, NAM$C_BLN, NAM);
2           FAB [FAB$B_BID] = FAB$C_BID;
2           FAB [FAB$B_BLN] = FAB$C_BLN;
2           NAM [NAM$B_BID] = NAM$C_BID;
2           NAM [NAM$B_BLN] = NAM$C_BLN;
2           FAB [FAB$L_NAM] = NAM;
2       !+
2       ! Set up common default value for FILE and DEFAULTFILE if needed
2       !-
2           UNIT = .OPEN [OPEN$K_UNIT];
2           IF .OPEN [OPEN$K_NAME] EQLA 0 OR
2               .OPEN [OPEN$K_DEFAULTF] EQLA 0
2           THEN
3               BEGIN
3               DEF_NAME [0] = %C'F';
3               DEF_NAME [1] = %C'O';
3               DEF_NAME [2] = %C'R';
3               DEF_NAME [3] = ((.UNIT/100) MOD 10) + %C'0';
3               DEF_NAME [4] = ((.UNIT/10) MOD 10) + %C'0';
3               DEF_NAME [5] = ((.UNIT) MOD 10) + %C'0';
3               DEF_NAME [6] = %C'.';
3               DEF_NAME [7] = %C'D';
3               DEF_NAME [8] = %C'A';
3               DEF_NAME [9] = %C'T';
3               END;
2
2       !+
2       ! Set up DEFAULTFILE name
2       !-
2
2           NAM_DSC = .OPEN [OPEN$K_DEFAULTF];
2
3           IF (.NAM_DSC NEQ 0)
2           THEN
3               BEGIN
3       !+
3       ! Default file name was specified.  Check for proper length then
3       ! use it.
3       !-
4               IF ((.NAM_DSC [DSC$W_LENGTH] GTRU 255) OR (.NAM_DSC [DSC$W_LENGTH] EQL 0))
3               THEN
3                   FOR$$SIG_NO_LUB (FOR$K_FILNAMSPE, .UNIT);
3
3               FAB [FAB$B_DNS] = .NAM_DSC [DSC$W_LENGTH];
3               FAB [FAB$L_DNA] = .NAM_DSC [DSC$A_POINTER];
3               END
2           ELSE
3               BEGIN
```

```
3 !+
3 ! DEFAULTFILE not specified, use name of FORnnn.DAT
3 !-
3               FAB [FAB$B_DNS] = %CHARCOUNT ('FORnnn.DAT');
3               FAB [FAB$L_DNA] = DEF_NAME;
2               END;
2
2 !+
2 ! Set up file name
2 !-
2          NAM_DSC = .OPEN [OPEN$K_NAME];
2
3          IF (.NAM_DSC NEQ 0)
2          THEN
3               BEGIN
3 !+
3 ! File name was specified.  Check for proper length then
3 ! use it.
3 !-
3
4               IF ((.NAM_DSC [DSC$W_LENGTH] GTRU 255) OR (.NAM_DSC [DSC$W_LENGTH] EQL 0))
3               THEN
3                   FOR$$SIG_NO_LUB (FOR$K_FILNAMSPE, .UNIT);
3
3               FAB [FAB$B_FNS] = .NAM_DSC [DSC$W_LENGTH];
3               FAB [FAB$L_FNA] = .NAM_DSC [DSC$A_POINTER];
3               END
2          ELSE
3               BEGIN
3 !+
3 ! File name not specified, use name of FORnnn which may be
3 ! a logical name.
3 !-
3               FAB [FAB$B_FNS] = %CHARCOUNT ('FORnnn');
3               FAB [FAB$L_FNA] = DEF_NAME;
2               END;
2
2 !+
2 ! Set up resultant name string
2 !-
2          NAM [NAM$B_ESS] = NAM [NAM$B_RSS] = NAM$C_MAXRSS;
2          NAM [NAM$L_ESA] = NAM [NAM$L_RSA] = RES_NAME;
2 !+
2 ! Parse and search for the file to get the resultant name
2 !-
2          RMS_STATUS = $PARSE (FAB = FAB);
2
2          IF (.RMS_STATUS) THEN $SEARCH (FAB = FAB) ELSE FOR$$SIG_NO_LUB (FOR$K_FILNAMSPE, .UNIT, FAB);
2
2 !+
2 ! Specifically forbid wildcards in file name.
2 !-
2
3          IF (.NAM [NAM$V_WILDCARD])
2          THEN
3               BEGIN
3               NAM [NAM$L_ESA] = 0;                         ! Invalidate result string
```

```
            NAM [NAM$L_RSA] = 0;
            FAB [FAB$L_STS] = 0;                        ! Invalidate statuses
            FAB [FAB$L_STV] = 0;
            FOR$$SIG_NO_LUB (FOR$K_FILNAMSPE, .UNIT, FAB);
            END;

!+
! See if the resultant name matches that stored in the LUB
! or if the name was not given and the unit is open.
!-
        RES_LEN = MAX (.NAM [NAM$B_RSL], .NAM [NAM$B_ESL]);

        IF ((CH$EQL (.RES_LEN, RES_NAME, .CCB [LUB$B_RSL], .CCB [LUB$A_RSN], %C' '))        !
            OR ((.OPEN [OPEN$K_NAME] EQL 0) AND .CCB [LUB$V_OPENED]))
        THEN
            BEGIN
!+
! Names match, change BLANK= value only.
!-

            CASE .OPEN [OPEN$K_BLANK] FROM 0 TO OPEN$K_BLK_NUL OF
                SET

                [0] :
                ;                                       ! Make no changes

                [OPEN$K_BLK_ZER] :                      ! BLANK='ZERO'
                    CCB [LUB$V_NULLBLNK] = 0;

                [OPEN$K_BLK_NUL] :                      ! BLANK='NULL'
                    CCB [LUB$V_NULLBLNK] = 1;

                [OUTRANGE] :
                    FOR$$SIG_NO_LUB (FOR$K_INVARGFOR, .UNIT, FAB);
                TES;

!+
! BLANK= set, now pop the LUB/RAB/ISB and return to FOR$OPEN
!-
            FOR$$CB_POP ();
            .L_UNWIND_ACTION = FOR$K_UNWINDNOP;
            RETURN 1;                                   ! No more OPEN processing needed
            END
        ELSE
            BEGIN
!+
! File names do not match; close current file, open new one.
!-

            IF NOT FOR$$CLOSE_FILE () THEN FOR$$SIG_NO_LUB (FOR$K_CLOERR, .UNIT, FAB);

            FOR$$CB_POP ();
            .L_UNWIND_ACTION = FOR$K_UNWINDNOP;
!+
! Now, try to initiate re-opening of this unit
!-
```

```
;  937          0998  3                FOR$$CB_PUSH (.UNIT, LUB$K_LUN_MIN);
;  938          0999  3                .L_UNWIND_ACTION = FOR$K_UNWINDPOP;
;  939          1000  3
;  940          1001  4                IF ((.CCB [LUB$V_OPENED]) OR (.CCB [LUB$V_DEALLOC]))
;  941          1002  3                THEN
;  942          1003  3                    FOR$$SIGNAL_STO (FOR$K_RECIO_OPE);
;  943          1004  3
;  944          1005  2                END;
;  945          1006  2
;  946          1007  2            RETURN 0;                                            ! Continue OPEN processing
;  947          1008  1            END;                                                 ! of routine OPEN_ON_CONNECTED


                                                                                        .EXTRN  SYS$PARSE, SYS$SEARCH

                                                        01FC 00000 OPEN_ON_CONNECTED:
                                                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8                         ; 0770
                                     58 00000000G  00   9E 00002                        MOVAB   FOR$$CB_POP, R8                                   ;
                                     57 00000000G  00   9E 00009                        MOVAB   FOR$$SIG_NO_LUB, R7                               ;
                                     5E      FE44  CE   9E 00010                        MOVAB   -444(SP), SP                                      ;
    0050    8F             00        6E            00   2C 00015                        MOVC5   #0, (SP), #0, #80, FAB                            ; 0835
                                               B0  AD      0001C                                                                                  ;
    0060    8F             00        6E            00   2C 0001E                        MOVC5   #0, (SP), #0, #96, NAM                            ; 0836
                                             FF50  CD      00025                                                                                  ;
                        B0      AD           5003  8F   B0 00028                        MOVW    #20483, FAB                                       ; 0837
                      FF50      CD           6002  8F   B0 0002E                        MOVW    #24578, NAM                                       ; 0839
                        D8      AD           FF50  CD   9E 00035                        MOVAB   NAM, FAB+40                                       ; 0841
                                54             04  AC   D0 0003B                        MOVL    OPEN, R4                                          ; 0845
                                55             04  A4   D0 0003F                        MOVL    4(R4), UNIT                                       ;
                                                   56   D4 00043                        CLRL    R6                                                ; 0846
                                               38  A4   D5 00045                        TSTL    56(R4)                                            ;
                                                   04   12 00048                        BNEQ    1$                                                ;
                                                   56   D6 0004A                        INCL    R6                                                ;
                                                   05   11 0004C                        BRB     2$                                                ;
                                               68  A4   D5 0004E  1$:                   TSTL    104(R4)                                           ; 0847
                                                   4B   12 00051                        BNEQ    3$                                                ;
                                6E           4F46  8F   B0 00053  2$:                   MOVW    #20294, DEF_NAME                                  ; 0850
                        02      AE             52  8F   90 00058                        MOVB    #82, DEF_NAME+2                                   ; 0852
                   52           55       00000064  8F   C7 0005D                        DIVL3   #100, UNIT, R2                                    ; 0853
     7E            00           52                 01   7A 00065                        EMUL    #1, R2, #0, -(SP)                                 ;
     52            52           8E                 0A   7B 0006A                        EDIV    #10, (SP)+, R2, R2                                ;
              03   AE           52                 30   81 0006F                        ADDB3   #48, R2, DEF_NAME+3                               ;
                   52           55                 0A   C7 00074                        DIVL3   #10, UNIT, R2                                     ; 0854
     7E            00           52                 01   7A 00078                        EMUL    #1, R2, #0, -(SP)                                 ;
     52            52           8E                 0A   7B 0007D                        EDIV    #10, (SP)+, R2, R2                                ;
              04   AE           52                 30   81 00082                        ADDB3   #48, R2, DEF_NAME+4                               ;
     7E            00           55                 01   7A 00087                        EMUL    #1, UNIT, #0, -(SP)                               ; 0855
     50            50           8E                 0A   7B 0008C                        EDIV    #10, (SP)+, R0, R0                                ;
              05   AE           50                 30   81 00091                        ADDB3   #48, R0, DEF_NAME+5                               ;
                        06      AE       5441442E  8F   D0 00096                        MOVL    #1413563438, DEF_NAME+6                           ; 0856
                                52             68  A4   D0 0009E  3$:                   MOVL    104(R4), NAM_DSC                                  ; 0866
                                                   1D   13 000A2                        BEQL    6$                                                ; 0868
                      00FF      8F                 62   B1 000A4                        CMPW    (NAM_DSC), #255                                   ; 0875
                                                   04   1A 000A9                        BGTRU   4$                                                ;
                                                   62   B5 000AB                        TSTW    (NAM_DSC)                                         ;
                                                   07   12 000AD                        BNEQ    5$                                                ;
```

```
                                    55  DD 000AF  4$:   PUSHL   UNIT                                     ; 0877
                                    2B  DD 000B1        PUSHL   #43                                      ;
                      67            02  FB 000B3        CALLS   #2, FOR$$SIG_NO_LUB                      ;
                 E5   AD            62  90 000B6  5$:   MOVB    (NAM_DSC), FAB+53                        ; 0879
                 E0   AD      04    A2  D0 000BA        MOVL    4(NAM_DSC), FAB+48                       ; 0880
                                    08  11 000BF        BRB     7$                                       ; 0868
                 E5   AD            0A  90 000C1  6$:   MOVB    #10, FAB+53                              ; 0887
                 E0   AD            6E  9E 000C5        MOVAB   DEF_NAME, FAB+48                         ; 0888
                      52      38    A4  D0 000C9  7$:   MOVL    56(R4), NAM_DSC                          ; 0894
                                    1D  13 000CD        BEQL    10$                                      ; 0896
               00FF   8F            62  B1 000CF        CMPW    (NAM_DSC), #255                          ; 0904
                                    04  1A 000D4        BGTRU   8$                                       ;
                                    62  B5 000D6        TSTW    (NAM_DSC)                                ;
                                    07  12 000D8        BNEQ    9$                                       ;
                                    55  DD 000DA  8$:   PUSHL   UNIT                                     ; 0906
                                    2B  DD 000DC        PUSHL   #43                                      ;
                      67            02  FB 000DE        CALLS   #2, FOR$$SIG_NO_LUB                      ;
                 E4   AD            62  90 000E1  9$:   MOVB    (NAM_DSC), FAB+52                        ; 0908
                 DC   AD      04    A2  D0 000E5        MOVL    4(NAM_DSC), FAB+44                       ; 0909
                                    08  11 000EA        BRB     11$                                      ; 0896
                 E4   AD            06  90 000EC  10$:  MOVB    #6, FAB+52                               ; 0917
                 DC   AD            6E  9E 000F0        MOVAB   DEF_NAME, FAB+44                         ; 0918
               FF52   CD            01  8E 000F4  11$:  MNEGB   #1, NAM+2                                ; 0924
               FF5A   CD            01  8E 000F9        MNEGB   #1, NAM+10                               ;
                      50      0C    AE  9E 000FE        MOVAB   RES_NAME, R0                             ; 0925
               FF54   CD            50  D0 00102        MOVL    R0, NAM+4                                ;
               FF5C   CD            50  D0 00107        MOVL    R0, NAM+12                               ;
                              B0    AD  9F 0010C        PUSHAB  FAB                                      ; 0929
          00000000G   00            01  FB 0010F        CALLS   #1, SYS$PARSE                            ;
                      0C            50  E9 00116        BLBC    RMS_STATUS, 12$                          ; 0931
                              B0    AD  9F 00119        PUSHAB  FAB                                      ;
          00000000G   00            01  FB 0011C        CALLS   #1, SYS$SEARCH                           ;
                                    0A  11 00123        BRB     13$                                      ;
                              B0    AD  9F 00125  12$:  PUSHAB  FAB                                      ;
                                    55  DD 00128        PUSHL   UNIT                                     ;
                                    2B  DD 0012A        PUSHL   #43                                      ;
                      67            03  FB 0012C        CALLS   #3, FOR$$SIG_NO_LUB                      ;
                      15      85    AD  E9 0012F  13$:  BLBC    NAM+53, 14$                              ; 0937
                            FF5C    CD  D4 00133        CLRL    NAM+12                                   ; 0940
                            FF54    CD  D4 00137        CLRL    NAM+4                                    ; 0941
                              B8    AD  7C 0013B        CLRQ    FAB+8                                    ; 0942
                              B0    AD  9F 0013E        PUSHAB  FAB                                      ; 0944
                                    55  DD 00141        PUSHL   UNIT                                     ;
                                    2B  DD 00143        PUSHL   #43                                      ;
                      67            03  FB 00145        CALLS   #3, FOR$$SIG_NO_LUB                      ;
                      50    FF53    CD  9A 00148  14$:  MOVZBL  NAM+3, R0                                ; 0951
                      50    FF5B    CD  91 0014D        CMPB    NAM+11, R0                               ;
                                    05  1B 00152        BLEQU   15$                                      ;
                      50    FF5B    CD  9A 00154        MOVZBL  NAM+11, R0                               ;
                      51            50  D0 00159  15$:  MOVL    R0, RES_LEN                              ;
                      50      F7    AB  9A 0015C        MOVZBL  -9(CCB), R0                              ; 0953
   50         20      0C      AE    51  2D 00160        CMPC5   RES_LEN, RES_NAME, #32, R0, @-8(CCB)     ;
                              F8    BB     00166                                                         ;
                                    07  13 00168        BEQL    16$                                      ;
                      31            56  E9 0016A        BLBC    R6, 21$                                  ; 0954
                      2D      FC    AB  E9 0016D        BLBC    -4(CCB), 21$                             ;
              02      00      60    A4  CF 00171  16$:  CASEL   96(R4), #0, #2                           ; 0961
              0019       0012        001E    00176 17$:     .WORD   20$-17$,-                                   ;
                                                                    18$-17$,-                                   ;
                                                                    19$-17$                                     ;
                                B0    AD  9F 0017C          PUSHAB  FAB                                         ; 0974
                                      55  DD 0017F          PUSHL   UNIT                                        ;
                                      30  DD 00181          PUSHL   #48                                         ;
                         67           03  FB 00183          CALLS   #3, FOR$$SIG_NO_LUB                         ;
                                      0C  11 00186          BRB     20$                                         ;
                   FF    AB     40    8F  8A 00188 18$:     BICB2   #64, -1(CCB)                                ; 0968
                                      05  11 0018D          BRB     20$                                         ;
                   FF    AB     40    8F  88 0018F 19$:     BISB2   #64, -1(CCB)                                ; 0971
                                      68  16 00194 20$:     JSB     FOR$$CB_POP                                 ; 0980
                   08    BC           01  D0 00196          MOVL    #1, @L_UNWIND_ACTION                        ; 0981
                         50           01  D0 0019A          MOVL    #1, R0                                      ; 0982
                                      04 0019D              RET                                                 ;
            00000000G    00           00  FB 0019E 21$:     CALLS   #0, FOR$$CLOSE_FILE                         ; 0990
                         0A           50  E8 001A5          BLBS    R0, 22$                                     ;
                                B0    AD  9F 001A8          PUSHAB  FAB                                         ;
                                      55  DD 001AB          PUSHL   UNIT                                        ;
                                      1C  DD 001AD          PUSHL   #28                                         ;
                         67           03  FB 001AF          CALLS   #3, FOR$$SIG_NO_LUB                         ;
                                      68  16 001B2 22$:     JSB     FOR$$CB_POP                                 ; 0992
                   08    BC           01  D0 001B4          MOVL    #1, @L_UNWIND_ACTION                        ; 0993
                                      50  D4 001B8          CLRL    R0                                          ; 0998
                         52           55  D0 001BA          MOVL    UNIT, R2                                    ;
                            00000000G 00  16 001BD          JSB     FOR$$CB_PUSH                                ;
                                08    BC  D4 001C3          CLRL    @L_UNWIND_ACTION                            ; 0999
                         05     FC    AB  E8 001C6          BLBS    -4(CCB), 23$                                ; 1001
              09   FF    AB           04  E1 001CA          BBC     #4, -1(CCB), 24$                            ;
                                      28  DD 001CF 23$:     PUSHL   #40                                         ; 1003
            00000000G    00           01  FB 001D1          CALLS   #1, FOR$$SIGNAL_STO                         ;
                                      50  D4 001D8 24$:     CLRL    R0                                          ; 1007
                                      04 001DA              RET                                                 ; 1008

; Routine Size:  475 bytes,    Routine Base:  _FOR$CODE + 029C


;  948        1009  1 END                                     ! End of FOR$OPEN module
;  949        1010  1
;  950        1011  0 ELUDOM




;                              PSECT SUMMARY
;
;       Name                  Bytes                      Attributes
;
;   _FOR$CODE                      1143  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         32          0            581         00:01.1
;   _$255$DUA28:[FORRTL.OBJ]FORLIB.L32;1        711        223         31             52         00:00.5
;   _$255$DUA28:[FORRTL.OBJ]RTLLIB.L32;1         36          0          0              8         00:00.1



;                                      COMMAND QUALIFIERS

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

; Size:           1143 code + 0 data bytes
; Run Time:          00:25.4
; Elapsed Time:      01:10.9
; Lines/CPU Min:       2392
; Lexemes/CPU-Min: 15555
; Memory Used:   233 pages
; Compilation Complete
```

0182 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

FORRAB
LIS

FORREADDF
LIS

FOROPNKEY
LIS

FORPAUSE
LIS

FORRANDOM
LIS

FOROPEN
LIS

FOROPENDE
LIS

FORREADDO
LIS