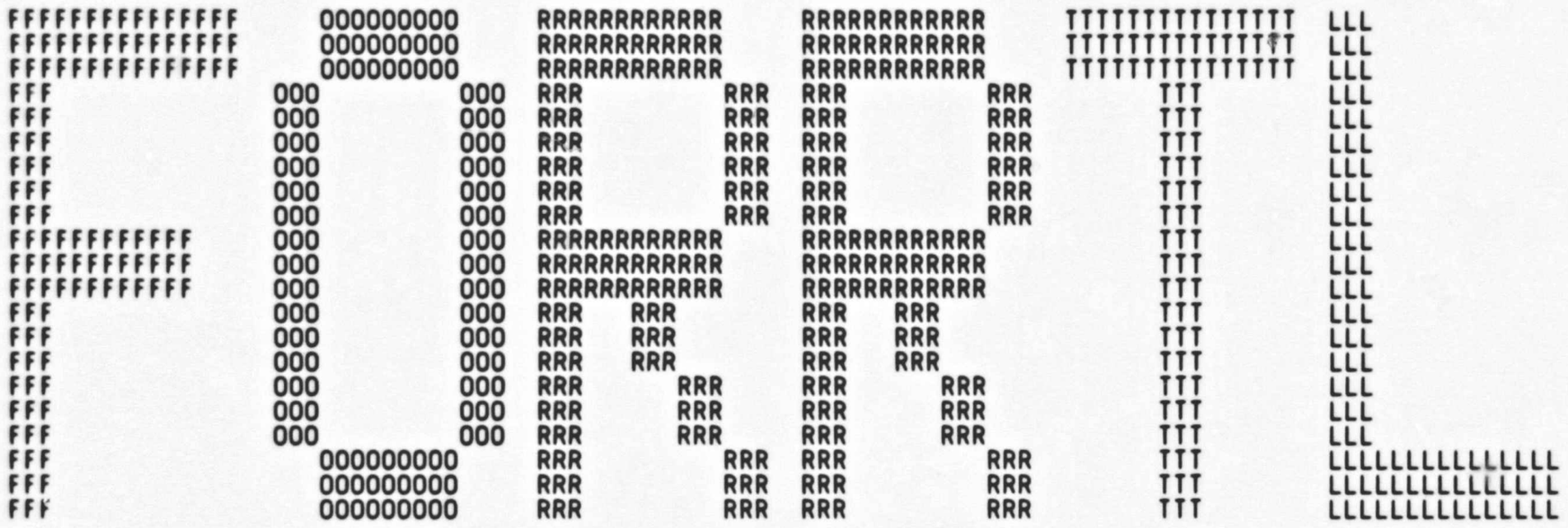
FORTRAN
LLLLLLLLLLLLLLL

FILEID**FOREXITHA

```
FFFFFFFFFF    000000    RRRRRRRR    EEEEEEEEEE  XX      XX    IIIIII    TTTTTTTTTT  HH      HH    AAAAAA
FFFFFFFFFF    000000    RRRRRRRR    EEEEEEEEEE  XX      XX    IIIIII    TTTTTTTTTT  HH      HH    AAAAAA
FF          00      00  RR      RR  EE          XX      XX      II        TT      HH      HH  AA      AA
FF          00      00  RR      RR  EE          XX      XX      II        TT      HH      HH  AA      AA
FF          00      00  RR      RR  EE            XX  XX        II        TT      HH      HH  AA      AA
FF          00      00  RR      RR  EE            XX  XX        II        TT      HH      HH  AA      AA
FFFFFFFF    00      00  RRRRRRRR    EEEEEEEE        XX          II        TT      HHHHHHHHHH  AA      AA
FFFFFFFF    00      00  RRRRRRRR    EEEEEEEE        XX          II        TT      HHHHHHHHHH  AA      AA
FF          00      00  RR  RR      EE              XX  XX      II        TT      HH      HH  AAAAAAAAAA
FF          00      00  RR  RR      EE              XX  XX      II        TT      HH      HH  AAAAAAAAAA
FF          00      00  RR    RR    EE          XX      XX      II        TT      HH      HH  AA      AA    ....
FF          00      00  RR    RR    EE          XX      XX      II        TT      HH      HH  AA      AA    ....
FF            000000    RR      RR  EEEEEEEEEE  XX      XX    IIIIII      TT      HH      HH  AA      AA    ....
FF            000000    RR      RR  EEEEEEEEEE  XX      XX    IIIIII      TT      HH      HH  AA      AA    ....

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
```

FOR

.....

```
;     1          0001  0 MODULE FOR$$EXIT_HANDL (%TITLE 'FORTRAN exit handler'
;     2          0002  0                   IDENT = '1-013'          ! File: FOREXITHA.B32  Edit: STAN1013
;     3          0003  0                   ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************************
;     7          0007  1 !*                                                                                    *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                          *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                           *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                            *
;    11          0011  1 !*                                                                                    *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED           *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE          *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER           *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY           *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY           *
;    17          0017  1 !*  TRANSFERRED.                                                                    *
;    18          0018  1 !*                                                                                    *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE           *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT           *
;    21          0021  1 !*  CORPORATION.                                                                    *
;    22          0022  1 !*                                                                                    *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS           *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                         *
;    25          0025  1 !*                                                                                    *
;    26          0026  1 !*                                                                                    *
;    27          0027  1 !**************************************************************************************
;    28          0028  1
;    29          0029  1 !++
;    30          0030  1 ! FACILITY: FORTRAN support library - Exit handler
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !
;    34          0034  1 !       This module is used when the image exits to do
;    35          0035  1 !       FORTRAN post processing.  It purges I/O buffers
;    36          0036  1 !       and closes files with proper disposition.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT: User access mode; mixture of AST level or not.
;    39          0039  1 !
;    40          0040  1 ! Author:       John Sauter, Creation date: 23-JAN-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original from FOROPEN.  JBS 23-JAN-1979
;    45          0045  1 ! 1-002 - Call OTS$$PURGE_IOBU to flush any "dirty" buffer.  JBS 24-JAN-1979
;    46          0046  1 ! 1-003 - Move call to OTS$$PURGE_IOBU to OTS$CLOSE_FILE.  JBS 24-JAN-1979
;    47          0047  1 ! 1-004 - Change linkage for OTS$PUSH_CCB to JSB_CB_PUSH and for
;    48          0048  1 !           OTS$POP_CCB to JSB_CB_POP.  JBS 25-JAN-1979
;    49          0049  1 ! 1-005 - Use two dollar signs for non-user entries.  JBS 26-JAN-1979
;    50          0050  1 ! 1-006 - Add OTS$$CLOSE_ALL.  JBS 04-JUN-1979
;    51          0051  1 ! 1-007 - Change to FORTRAN-specific exit handler.  JBS 16-AUG-1979
;    52          0052  1 ! 1-008 - Clear FOR$$L_XIT_LOCK when EXIT_HANDLER is called.  This
;    53          0053  1 !         allows exit handler I/O to be cleaned up properly.  SBL 29-Apr-1980
;    54          0054  1 ! 1-009 - Request page alignment for the OWN PSECT when invoking
;    55          0055  1 !         DECLARE_PSECTS, to ensure that EXIT_BLOCK will not occupy the
;    56          0056  1 !         same page as a user variable that is being WATCHed, and thus
;    57          0057  1 !         be unwriteable when $DCLEXH is called.  JAW 19-FEB-1981
```

```
;    58          0058  1 ! 1-010 - Declare LIB$STOP external.  SBL 30-Nov-1981
;    59          0059  1 ! 1-011 - Reflect changes of OTS$$ structures to FOR$$.  SBL 26-Oct-1982
;    60          0060  1 ! 1-012 - Reflect use of ISB$A_PREVIOUS_LUB instead of second longword in
;    61          0061  1 !         FOR$$AA_CUR_LUB.  SBL 2-Dec-1982
;    62          0062  1 ! 1-013 - Remove informational error. STAN 24-Jul-1984.
;    63          0063  1 !--
;    64          0064  1
```

```
;    66          0065  1 !
;    67          0066  1 ! PROLOGUE FILE:
;    68          0067  1 !
;    69          0068  1
;    70          0069  1 REQUIRE 'RTLIN:FORPROLOG';                      ! FOR$ definitions
;    71          0135  1
;    72          0136  1 !
;    73          0137  1 ! TABLE OF CONTENTS:
;    74          0138  1 !
;    75          0139  1
;    76          0140  1 FORWARD ROUTINE
;    77          0141  1     FOR$$DECL_EXITH : NOVALUE,                  ! Declare EXIT handler
;    78          0142  1     EXIT_HANDLER : NOVALUE,                     ! Exit Handler
;    79          0143  1     FOR$$CLOSE_ALL : NOVALUE;                   ! Close all files
;    80          0144  1
;    81          0145  1 !
;    82          0146  1 ! MACROS:
;    83          0147  1 !
;    84          0148  1 !       NONE
;    85          0149  1 !
;    86          0150  1 ! EQUATED SYMBOLS:
;    87          0151  1 !
;    88          0152  1 !       NONE
;    89          0153  1 !
;    90          0154  1 ! OWN STORAGE:
;    91          0155  1 !
;    92          0156  1
;    93          0157  1 GLOBAL
;    94          0158  1     FOR$$L_XIT_LOCK : INITIAL (0);              ! Clear if no handler linked yet
;    95          0159  1
;    96          0160  1 !
;    97          0161  1 ! EXTERNAL REFERENCES:
;    98          0162  1 !
;    99          0163  1
;   100          0164  1 !+
;   101          0165  1 ! The following structure is used for addressing FOR$$AA_LUB_TAB.
;   102          0166  1 ! It is similar to VECTOR, but offsets the index so that
;   103          0167  1 ! negative logical unit numbers can be used.
;   104          0168  1 !-
;   105          0169  1
;   106          0170  1 STRUCTURE
;   107          0171  1     FOR$$LUB_TAB_ST [I; N, LB, UNIT = 4, EXT = 0] =
;   108          0172  1         [N*UNIT]
;   109          0173  1         (FOR$$LUB_TAB_ST + ((I - LB)*UNIT))<0, %BPUNIT*UNIT, EXT>;
;   110          0174  1
;   111          0175  1 EXTERNAL
;   112          0176  1     FOR$$AA_LUB_TAB : VOLATILE FOR$$LUB_TAB_ST  ! Table of LUB addresses
;   113          0177  1         [-LUB$K_ILUN_MIN + LUB$K_LUN_MAX + 1, LUB$K_ILUN_MIN],
;   114          0178  1     FOR$$A_CUR_LUB;                             ! Current LUB
;   115          0179  1
;   116          0180  1 EXTERNAL ROUTINE
;   117          0181  1     FOR$$CB_FETCH : NOVALUE,                    ! Load register CCB, no push
;   118          0182  1     FOR$$CB_PUSH : JSB_CB_PUSH NOVALUE,         ! Load register CCB
;   119          0183  1     FOR$$CB_POP : JSB_CB_POP NOVALUE,           ! Done with register CCB
;   120          0184  1     FOR$$GET_VM,                                ! Get virtual memory
;   121          0185  1     FOR$$NEXT_LUN : NOVALUE,                    ! Get next LUN that might be open
;   122          0186  1     FOR$$CLOSE_FILE : CALL_CCB,                 ! Internal file closer
```

```
;  123          0187  1     FOR$$SIGNAL: NOVALUE;                    ! Signal non-fatal error
;  124          0188  1
```

```
;  126          0189  1 GLOBAL ROUTINE FOR$$DECL_EXITH                    ! Declare VMS EXIT handler
;  127          0190  1     : NOVALUE =                                    !
;  128          0191  1
;  129          0192  1 !++
;  130          0193  1 ! FUNCTIONAL DESCRIPTION:
;  131          0194  1 !
;  132          0195  1 !       Declares VMS EXIT handler for FORTRAN.
;  133          0196  1 !
;  134          0197  1 ! CALLING SEQUENCE:
;  135          0198  1 !
;  136          0199  1 !       IF (NOT .FOR$$L_XIT_LOCK) THEN FOR$$DECL_EXITH ()
;  137          0200  1 !
;  138          0201  1 ! FORMAL PARAMETERS:
;  139          0202  1 !
;  140          0203  1 !       NONE
;  141          0204  1 !
;  142          0205  1 ! IMPLICIT INPUTS:
;  143          0206  1 !
;  144          0207  1 !       NONE
;  145          0208  1 !
;  146          0209  1 ! IMPLICIT OUTPUTS:
;  147          0210  1 !
;  148          0211  1 !       NONE
;  149          0212  1 !
;  150          0213  1 ! COMPLETION CODES:
;  151          0214  1 !
;  152          0215  1 !       NONE
;  153          0216  1 !
;  154          0217  1 ! SIDE EFFECTS:
;  155          0218  1 !
;  156          0219  1 !       Declares VMS EXIT handler.
;  157          0220  1 !--
;  158          0221  1
;  159          0222  2     BEGIN
;  160          0223  2
;  161          0224  2
;  162          0225  2     !+
;  163          0226  2     ! Allocate and fill in the control descriptor block.
;  164          0227  2     !-
;  165          0228  2
;  166          0229  2     LOCAL
;  167          0230  2         EXITH_CONTROL_BLOCK: REF VECTOR [, LONG];
;  168          0231  2
;  169          0232  2     EXITH_CONTROL_BLOCK = FOR$$GET_VM (20);      ! 5 longwords
;  170          0233  2     EXITH_CONTROL_BLOCK [1] = EXIT_HANDLER;      ! Routine address
;  171          0234  2     EXITH_CONTROL_BLOCK [2] = 1;                 ! 1 additional longword
;  172          0235  2     EXITH_CONTROL_BLOCK [3] = EXITH_CONTROL_BLOCK [4]; ! Reason for exit
;  173          0236  2     $DCLEXH (DESBLK=EXITH_CONTROL_BLOCK [0]);    ! Assume success
;  174          0237  2     FOR$$L_XIT_LOCK = 1;                         ! Handler declared
;  175          0238  2
;  176          0239  2     RETURN
;  177          0240  1     END;


                                                .TITLE  FOR$$EXIT_HANDL FORTRAN exit handler
                                                .IDENT  \1-013\
```

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

                                00000000  00000 FOR$$L_XIT_LOCK::
                                                .LONG   0                                       ;

                                                .EXTRN  FOR$$AA_LUB_TAB
                                                .EXTRN  FOR$$A_CUR_LUB, FOR$$CB_FETCH
                                                .EXTRN  FOR$$CB_PUSH, FOR$$CB_POP
                                                .EXTRN  FOR$$GET_VM, FOR$$NEXT_LUN
                                                .EXTRN  FOR$$CLOSE_FILE
                                                .EXTRN  FOR$$SIGNAL, SYS$DCLEXH

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

                                     0000 00000         .ENTRY  FOR$$DECL_EXITH, Save nothing        ; 0189
                                  14   DD 00002         PUSHL   #20                                  ; 0232
                    00000000G   00    01   FB 00004    CALLS   #1, FOR$$GET_VM                      ;
                            04  A0      0000V CF  9E 0000B    MOVAB   EXIT_HANDLER, 4(EXITH_CONTROL_BLOCK)  ; 0233
                            08  A0            01  D0 00011    MOVL    #1, 8(EXITH_CONTROL_BLOCK)   ; 0234
                            0C  A0        10  A0  9E 00015    MOVAB   16(R0), 12(EXITH_CONTROL_BLOCK)  ; 0235
                                          50  DD 0001A        PUSHL   EXITH_CONTROL_BLOCK          ; 0236
                    00000000G   00        01  FB 0001C       CALLS   #1, SYS$DCLEXH               ;
                    00000000'   EF        01  D0 00023       MOVL    #1, FOR$$L_XIT_LOCK          ; 0237
                                              04 0002A       RET                                  ; 0240

; Routine Size:  43 bytes,    Routine Base:  _FOR$CODE + 0000

;  178          0241  1
```

```
;  180          0242  1 ROUTINE EXIT_HANDLER (                          ! Exit Handler
;  181          0243  1         EXIT_REASON                             ! Reason
;  182          0244  1     ) : NOVALUE =
;  183          0245  1
;  184          0246  1 !++
;  185          0247  1 ! FUNCTIONAL DESCRIPTION:
;  186          0248  1 !
;  187          0249  1 !       This is the exit handler for FORTRAN.  Its only function is to
;  188          0250  1 !       close all files.
;  189          0251  1 !
;  190          0252  1 !       Upon entering, it clears FOR$$L_XIT_LOCK.  This allows user
;  191          0253  1 !       exit handlers to have their I/O cleaned up.
;  192          0254  1 !
;  193          0255  1 ! FORMAL PARAMETERS:
;  194          0256  1 !
;  195          0257  1 !       EXIT_REASON.rl.r        not used
;  196          0258  1 !
;  197          0259  1 ! IMPLICIT INPUTS:
;  198          0260  1 !
;  199          0261  1 !       NONE
;  200          0262  1 !
;  201          0263  1 ! IMPLICIT OUTPUTS:
;  202          0264  1 !
;  203          0265  1 !       FOR$$L_XIT_LOCK is set to zero
;  204          0266  1 !
;  205          0267  1 ! ROUTINE VALUE:
;  206          0268  1 ! COMPLETION CODES:
;  207          0269  1 !
;  208          0270  1 !       NONE
;  209          0271  1 !
;  210          0272  1 ! SIDE EFFECTS:
;  211          0273  1 !
;  212          0274  1 !       Closes all files by calling OTS$$CLOSE_ALL.
;  213          0275  1 !--
;  214          0276  1
;  215          0277  2     BEGIN
;  216          0278  2     FOR$$L_XIT_LOCK = 0;                ! Clear exit handler interlock
;  217          0279  2     FOR$$CLOSE_ALL ();
;  218          0280  1     END;                                    ! of routine EXIT_HANDLER



                                0000 00000 EXIT_HANDLER:
                                           .WORD   Save nothing                          ; 0242
                    00000000'   EF  D4 00002       CLRL    FOR$$L_XIT_LOCK           ; 0278
              0000V CF          00  FB 00008       CALLS   #0, FOR$$CLOSE_ALL        ; 0279
                                04 0000D       RET                               ; 0280

; Routine Size:  14 bytes,    Routine Base:  _FOR$CODE + 002B
```

```
;  220          0281  1 ROUTINE FOR$$CLOSE_ALL : NOVALUE =                ! Close all files
;  221          0282  1
;  222          0283  1 !++
;  223          0284  1 ! FUNCTIONAL DESCRIPTION:
;  224          0285  1 !
;  225          0286  1 !       Find every existing LUB (with a linear search through the LUB
;  226          0287  1 !       table).  For each LUB, if the file is open, purge its I/O
;  227          0288  1 !       buffers and close it.  If the file has been marked for PRINT
;  228          0289  1 !       or DELETE, this will cause proper disposition of the file.
;  229          0290  1 !       RMS will close all open files at image exit, but it doesn't know
;  230          0291  1 !       about the above two DISPOSE conditions.  We couldn't set them at
;  231          0292  1 !       OPEN time, since the user is allowed to specify a different
;  232          0293  1 !       DISPOSE option at close time (with the CLOSE statement).
;  233          0294  1 !
;  234          0295  1 ! FORMAL PARAMETERS:
;  235          0296  1 !
;  236          0297  1 !       NONE
;  237          0298  1 !
;  238          0299  1 ! IMPLICIT INPUTS:
;  239          0300  1 !
;  240          0301  1 !       NONE
;  241          0302  1 !
;  242          0303  1 ! IMPLICIT OUTPUTS:
;  243          0304  1 !
;  244          0305  1 !       NONE
;  245          0306  1 !
;  246          0307  1 ! ROUTINE VALUE:
;  247          0308  1 ! COMPLETION CODES:
;  248          0309  1 !
;  249          0310  1 !       NONE
;  250          0311  1 !
;  251          0312  1 ! SIDE EFFECTS:
;  252          0313  1 !
;  253          0314  1 !       Closes all files.
;  254          0315  1 !       May signal FOR$_CLOERR.
;  255          0316  1 !--
;  256          0317  1
;  257          0318  2     BEGIN
;  258          0319  2
;  259          0320  2     GLOBAL REGISTER
;  260          0321  2         CCB = K_CCB_REG : REF $FOR$CCB_DECL;
;  261          0322  2
;  262          0323  2     LOCAL
;  263          0324  2         AST_STATUS,     ! Status from $SETAST
;  264          0325  2         FLAG,
;  265          0326  2         LUN;
;  266          0327  2
;  267          0328  2     !+
;  268          0329  2     ! Disable ASTs.
;  269          0330  2     !-
;  270          0331  2
;  271          0332  2     AST_STATUS = $SETAST (ENBFLG=0);
;  272          0333  2
;  273          0334  2     !+
;  274          0335  2     ! Scan through the table of LUNs, looking for allocated CCBs.
;  275          0336  2     !-
;  276          0337  2
```

```
;  277          0338  2     FLAG = 0;   ! Initialize loop
;  278          0339  2
;  279          0340  2     WHILE 1 DO
;  280          0341  3         BEGIN
;  281          0342  3
;  282          0343  3         !+
;  283          0344  3         ! Get next unit number that has an allocated CCB.
;  284          0345  3         !-
;  285          0346  3
;  286          0347  3         FOR$$NEXT_LUN (FLAG, LUN);
;  287          0348  3         IF NOT .FLAG
;  288          0349  3         THEN
;  289          0350  3             EXITLOOP;
;  290          0351  3
;  291          0352  3         !+
;  292          0353  3         ! Get the CCB address for this unit.
;  293          0354  3         !-
;  294          0355  3
;  295          0356  3         CCB = 0;                    ! Make BLISS understand that CCB is set.
;  296          0357  3         FOR$$CB_FETCH (.LUN);
;  297          0358  3         FOR$$A_CUR_LUB = .CCB;  ! Make this LUB the current one
;  298          0359  3
;  299          0360  3         !+
;  300          0361  3         ! We have an allocated CCB.  See if it has an open file.  If
;  301          0362  3         ! so, try to close it.  If the close fails, signal "close error",
;  302          0363  3         ! but with E severity so that execution continues.
;  303          0364  3         !-
;  304          0365  3
;  305          0366  3         IF .CCB [LUB$V_OPENED]
;  306          0367  3         THEN
;  307          0368  4             BEGIN
;  308          0369  4             IF NOT FOR$$CLOSE_FILE ()
;  309          0370  4             THEN
;  310          0371  5                 BEGIN
;  311          0372  5                 LOCAL
;  312          0373  5                     CLO_ERROR: BLOCK [4, BYTE]; ! Condition to signal
;  313          0374  5                 CLO_ERROR = FOR$_CLOERR;
;  314          0375  5                 CLO_ERROR [STS$V_SEVERITY] = STS$K_ERROR;
;  315          0376  5                 FOR$$SIGNAL (.CLO_ERROR);
;  316          0377  4                 END;
;  317          0378  3             END;
;  318          0379  3
;  319          0380  3         !+
;  320          0381  3         ! Zero the CCB address in the table of CCBs/LUBs.  This prevents
;  321          0382  3         ! anything else from happening to this incarnation of the file.
;  322          0383  3         !-
;  323          0384  3
;  324          0385  3         FOR$$AA_LUB_TAB [.LUN] = 0;
;  325          0386  2         END;
;  326          0387  2
;  327          0388  2     !+
;  328          0389  2     ! If ASTs were previously enabled, re-enable them.
;  329          0390  2     !-
;  330          0391  2
;  331          0392  2     IF .AST_STATUS EQL SS$_WASSET
;  332          0393  2     THEN
;  333          0394  2         $SETAST (ENBFLG=1);
```

```
;  334          0395  2
;  335          0396  2     RETURN;
;  336          0397  1     END;                                    ! of routine FOR$$CLOSE_ALL


                                                .EXTRN  SYS$SETAST

                                        080C 00000 FOR$$CLOSE_ALL:
                                                .WORD   Save R2,R3,R11                          ; 0281
                          53 00000000G  00  9E 00002            MOVAB   SYS$SETAST, R3          ;
                          5E            08  C2 00009            SUBL2   #8, SP                  ;
                                        7E  D4 0000C            CLRL    -(SP)                   ; 0332
                          63            01  FB 0000E            CALLS   #1, SYS$SETAST          ;
                          52            50  D0 00011            MOVL    R0, AST_STATUS          ;
                                  04    AE  D4 00014            CLRL    FLAG                    ; 0338
                                        5E  DD 00017 1$:        PUSHL   SP                      ; 0347
                                  08    AE  9F 00019            PUSHAB  FLAG                    ;
                    00000000G     00        02 FB 0001C         CALLS   #2, FOR$$NEXT_LUN       ;
                          41        04    AE  E9 00023            BLBC    FLAG, 3$                ; 0348
                                        5B  D4 00027            CLRL    CCB                     ; 0356
                                        6E  DD 00029            PUSHL   LUN                     ; 0357
                    00000000G     00        01 FB 0002B         CALLS   #1, FOR$$CB_FETCH       ;
                    00000000G     00        5B D0 00032         MOVL    CCB, FOR$$A_CUR_LUB     ; 0358
                          1F        FC  AB  E9 00039            BLBC    -4(CCB), 2$             ; 0366
                    00000000G     00        00 FB 0003D         CALLS   #0, FOR$$CLOSE_FILE     ; 0369
                          15            50  E8 00044            BLBS    R0, 2$                  ;
                                50 001880E4 8F  D0 00047            MOVL    #1605860, CLO_ERROR     ; 0374
          50          03                00  02 F0 0004E            INSV    #2, #0, #3, CLO_ERROR   ; 0375
                                        50  DD 00053            PUSHL   CLO_ERROR               ; 0376
                    00000000G     00        01 FB 00055         CALLS   #1, FOR$$SIGNAL         ;
                          50            6E  D0 0005C 2$:        MOVL    LUN, R0                 ; 0385
                              00000000G0040 D4 0005F            CLRL    FOR$$AA_LUB_TAB+32[R0]  ;
                                        AF  11 00066            BRB     1$                      ; 0340
                          09            52  D1 00068 3$:        CMPL    AST_STATUS, #9          ; 0392
                                        05  12 0006B            BNEQ    4$                      ;
                                        01  DD 0006D            PUSHL   #1                      ; 0394
                          63            01  FB 0006F            CALLS   #1, SYS$SETAST          ;
                                        04 00072 4$:            RET                             ; 0397

; Routine Size:  115 bytes,    Routine Base:  _FOR$CODE + 0039

;  337          0398  1 END                                     ! End of FOR$$EXIT_HANDL module
;  338          0399  1
;  339          0400  0 ELUDOM


;                                PSECT SUMMARY
;
;      Name                      Bytes                         Attributes
;
;  _FOR$DATA                       4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
```

```
;  _FOR$CODE                    172  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          7         0         581        00:01.0
;  _$255$DUA28:[FORRTL.OBJ]FORLIB.L32;1              711        184        25          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$:FOREXITHA/OBJ=OBJ$:FOREXITHA MSRC$:FOREXITHA/UPDATE=(ENH$:FOREXITHA
;       )

; Size:          172 code + 4 data bytes
; Run Time:        00:06.9
; Elapsed Time:    00:21.9
; Lines/CPU Min:    3488
; Lexemes/CPU-Min:  9723
; Memory Used:  90 pages
; Compilation Complete
```

FORFIND
LIS

FORFMTINT
LIS

FOREXITHA
LIS

FORENODEF
LIS

FORENCOMF
LIS

FORIDATE
LIS

FORENDFIL
LIS

FORERRSNS
LIS

FOREXIT
LIS

FORERROR
LIS

FORFMTCP
LIS

FORENCOMO
LIS

FORTNIDES
LIS