



```

FFFFFFFFF 000000 RRRRRRRR IIIIII 000000 EEEEEEEEE NN NN DDDDDDD
FFFFFFFFF 000000 RRRRRRRR IIIIII 000000 EEEEEEEEE NN NN DDDDDDD
FF 00 00 RR RR 00 00 EE NN NN DD DD
FF 00 00 RR RR 00 00 EE NN NN DD DD
FF 00 00 RR RR 00 00 EE NNNN NN DD DD
FF 00 00 RR RR 00 00 EE NNNN NN DD DD
FFFFFFFFF 00 00 RRRRRRRR IIIIII 00 00 EEEEEEEE NN NN DD DD
FFFFFFFFF 00 00 RRRRRRRR IIIIII 00 00 EEEEEEEE NN NN DD DD
FF 00 00 RR RR IIIIII 00 00 EE NN NNNN DD DD
FF 00 00 RR RR IIIIII 00 00 EE NN NNNN DD DD
FF 00 00 RR RR IIIIII 00 00 EE NN NN DD DD
FF 00 00 RR RR IIIIII 00 00 EE NN NN DD DD
FF 000000 RR RR IIIIII 000000 EEEEEEEEE NN NN DDDDDDD
FF 000000 RR RR IIIIII 000000 EEEEEEEEE NN NN DDDDDDD

```

```

LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSS
LL II SSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLL IIIIII SSSSSSS
LLLLLLLLL IIIIII SSSSSSS

```

FO  
Sy  
FO  
SY  
Ph  
--  
\_F  
Ph  
--  
In  
Co  
Pa  
Sy  
Pa  
Sy  
Ps  
Cr  
As  
Th  
16  
Th  
13  
2  
Ma  
\_S  
13  
Th  
MA

```

: 1 0001 0 MODULE FOR$IO_END (%TITLE 'FORTRAN End I/O statement'
: 2 0002 0 IDENT = '1-011' ! File: FORIOEND.B32 Edit: SBL1011
: 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 - user callable
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module terminates a FORTRAN I/O statement, writes
35 0035 1 last record if output, and pops up the I/O system to
36 0036 1 a previously active I/O statement if any.
37 0037 1
38 0038 1 ENVIRONMENT: User access mode; mixture of AST level or not
39 0039 1
40 0040 1 AUTHOR: Thomas N. Hastings, CREATION DATE: 03-Mar-77
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 Thomas N. Hastings, 05-Mar-77 : VERSION 01
45 0045 1 [Previous edit history removed. SBL 29-Sept-1982]
46 0046 1 1-011 - Change OTS$$ data structure references to FOR$$ SBL 29-Sep-1982
47 0047 1 --
48 0048 1

```

```
50 0049 1 |  
51 0050 1 | PROLOGUE FILE:  
52 0051 1 |  
53 0052 1 |  
54 0053 1 REQUIRE 'RTLIN:FORPROLOG';           ! FORTRAN declarations  
55 0119 1 |  
56 0120 1 |  
57 0121 1 | TABLE OF CONTENTS:  
58 0122 1 |  
59 0123 1 |  
60 0124 1 FORWARD ROUTINE  
61 0125 1   FOR$IO_END;                       ! End I/O statement  
62 0126 1 |  
63 0127 1 |  
64 0128 1 | EQUATED SYMBOLS:  
65 0129 1 |  
66 0130 1 |     NONE  
67 0131 1 |  
68 0132 1 | OWN STORAGE:  
69 0133 1 |  
70 0134 1 |     NONE  
71 0135 1 |  
72 0136 1 | EXTERNAL REFERENCES:  
73 0137 1 |  
74 0138 1 |  
75 0139 1 | EXTERNAL  
76 0140 1 |   FOR$$A_CUR_LUB : VOLATILE,                ! Pointer to current LUB/ISB/RAB  
77 0141 1 |   FOR$$A_UDF_PR9 : VECTOR;                 ! PIC array of user data  
78 0142 1 |                                           ! formatter (UDF) level of abstraction.  
79 0143 1 |  
80 0144 1 | EXTERNAL ROUTINE  
81 0145 1 |   FOR$$ERR_ENDHND,                          ! error condition handler for END= and ERR=  
82 0146 1 |   FOR$$CB_POP : JSB CB_POP NOVALUE,        ! Pop entire I/O system back to previous LUB/ISB/RAB  
83 0147 1 |   FOR$$SIGNAL : NOVALUE;                   ! Signal an error with a small error number  
84 0148 1 |
```

```

: 86      0149 1 GLOBAL ROUTINE FOR$IO_END =
: 87      0150 1
: 88      0151 1
: 89      0152 1  ++
: 90      0153 1  FUNCTIONAL DESCRIPTION:
: 91      0154 1
: 92      0155 1  CALLING SEQUENCE:
: 93      0156 1
: 94      0157 1          iostat.wl.v = FOR$IO_END ()
: 95      0158 1
: 96      0159 1  FORMAL PARAMETERS:
: 97      0160 1
: 98      0161 1          NONE
: 99      0162 1
: 100     0163 1  IMPLICIT INPUTS:
: 101     0164 1
: 102     0165 1          FOR$$A_CUR_LUB          Adr. of current logical unit
: 103     0166 1          block (LUB). Used to setup ISB
: 104     0167 1          to get current I/O statement type
: 105     0168 1          code.
: 106     0169 1          ISB$B_STTM_TYPE        I/O statement type code - index to
: 107     0170 1          dispatch table entry.
: 108     0171 1          FOR$AA_UDF_PR1         Array of user data formatters
: 109     0172 1          (UDF level of abstraction).
: 110     0173 1          ISB$B_ERR_NO          Last continuable error to occur in the state-
: 111     0174 1          ment or 0. SIGNAL if non-zero!
: 112     0175 1          format array or 0 if none.
: 113     0176 1
: 114     0177 1  IMPLICIT OUTPUTS:
: 115     0178 1
: 116     0179 1          FOR$$A_CUR_LUB          Adr. of pushed down LUB/ISB/RAB or 0
: 117     0180 1          if no unit pushed down (usual)
: 118     0181 1
: 119     0182 1  ROUTINE VALUE:
: 120     0183 1
: 121     0184 1          An IOSTAT small integer FORTRAN error number, either 0 if no
: 122     0185 1          error or the number of whatever continuable error last occurred
: 123     0186 1          in the I/O statement.
: 124     0187 1
: 125     0188 1  SIDE EFFECTS:
: 126     0189 1
: 127     0190 1          If an error occurs, it is SIGNALed unless an ERR=
: 128     0191 1          transfer parameter was specified when the I/O statement
: 129     0192 1          initialization call was made (see module FOR$IO BEG,
: 130     0193 1          entry points FOR$(READ,WRITE) {SF,SO,SU,DF,DO,DU,SL} or
: 131     0194 1          FOR$(DECODE,ENCODE) {MF,MO}), in which case control is
: 132     0195 1          transferred to the specified address (after stack
: 133     0196 1          unwind).
: 134     0197 1
: 135     0198 1  --
: 136     0199 1
: 137     0200 2  BEGIN
: 138     0201 2
: 139     0202 2  GLOBAL REGISTER
: 140     0203 2          CCB = K_CCB_REG : REF $FOR$CCB_DECL;
: 141     0204 2
: 142     0205 2  LOCAL          ! Declare locals for each item to be passed to handler

```

```

: 143      0206      2
: 144      0207      2
: 145      0208      2
: 146      0209      2
: 147      0210      2
: 148      0211      2
: 149      0212      2
: 150      0213      2
: 151      0214      2
: 152      0215      2
: 153      0216      2
: 154      0217      2
: 155      0218      2
: 156      0219      2
: 157      0220      2
: 158      0221      2
: 159      0222      2
: 160      0223      2
: 161      0224      2
: 162      0225      2
: 163      0226      2
: 164      0227      2
: 165      0228      2
: 166      0229      2
: 167      0230      2
: 168      0231      2
: 169      0232      2
: 170      0233      2
: 171      0234      2
: 172      0235      2
: 173      0236      2
: 174      0237      2
: 175      0238      2
: 176      0239      2
: 177      0240      2
: 178      0241      2
: 179      0242      2
: 180      0243      2
: 181      0244      2
: 182      0245      2
: 183      0246      2
: 184      0247      2
: 185      0248      2
: 186      0249      2
: 187      0250      2
: 188      0251      2
: 189      0252      2
: 190      0253      2
: 191      0254      2
: 192      0255      2
: 193      0256      2
: 194      0257      2
: 195      0258      2
: 196      0259      2
: 197      0260      2
: 198      0261      2
: 199      0262      2

      L_UNWIND_ACTION : VOLATILE,
      A_ERR_ADR : VOLATILE,
      A_END_ADR : VOLATILE,
      L_INCR_DEPTH : VOLATILE,
      USER_FRAME : REF BLOCK [, BYTE],
      IOSTAT;

      ! Needed since can only pass address, not contents
      ! UNWIND action code.
      ! ERR= user address
      ! END= user address
      ! additional frames between establisher and user (0)
      ! User's call frame
      ! Local copy of ISB$B_ERR_NO

ENABLE
      FOR$$ERR_ENDHND (L_UNWIND_ACTION, A_ERR_ADR, A_END_ADR, L_INCR_DEPTH);
      ! Establish error conditon handler
      ! Pass UNWIND action code.

      ! Pass ERR= and END= user addresses or 0
      ! and the number of frames between the activator and the user (0)

CCB = .FOR$$A_CUR_LUB;

!+
! Setup LOCAL storage to be passed to error handler in case of a signal
! Indicate that UNWIND action is to pop current LUB/ISB/RAB if error
!-

L_UNWIND_ACTION = FOR$K_UNWINDPOP;
A_ERR_ADR = .CCB [ISB$A_ERR_EQUAL];
A_END_ADR = .CCB [ISB$A_END_EQUAL];
L_INCR_DEPTH = 0;

!+
! Restore user's handler in the frame, if any.
!-
USER_FRAME = .CCB [ISB$A_USER_FP];
USER_FRAME [SF$A_HANDLER] = .CCB [ISB$A_USR_HANDL];

!+
! Call appropriate UDF termination routine
! Any errors will be signaled.
!-

JSB_UDF9 (FOR$$AA_UDF_PR9 + .FOR$$AA_UDF_PR9 [CCB [ISB$B_STTM_TYPE] -
      ISB$K_FORSTTYLO + 1]);

!+
! If a continuable error occured on the I/O statement (ISB$B_ERR_NO
! is non-zero) then SIGNAL the error. The signalling is delayed until
! statement end time so the record is positioned correctly.
!-

IOSTAT = .CCB [ISB$B_ERR_NO];

IF .IOSTAT NEQU 0 THEN FOR$$SIGNAL (.IOSTAT);

! Pop entire I/O system back to previous LUB/ISB/RAB if there was one when this
! I/O statement began (see FIOBEG) or indicate that I/O system is not.
! currently processing any I/O statement (usual)
! LUB for (old) current unit is saved for next I/O statement
!-

FOR$$CB_POP ();
RETURN .IOSTAT;
      ! Return IOSTAT error value
```

: 200            0263 1    END;

! End of routine

```

083C 00000
53 00000000G 00 9E 00002
5E          0C C2 00009
           04 AE 7C 0000E
           0C AE D4 00011
6D          0049 CF DE 00014
5B 00000000G 00 D0 00019
           0C AE D4 00020
08 AE FF74 CB D0 00023
04 AE FF78 CB D0 00029
           6E D4 0002F
50 FF4C CB D0 00031
60 FF44 CB D0 00036
50 FF71 CB 9A 0003B
50          6340 D0 00040
           6340 16 00044
52 FF70 CB 9A 00047
           09 13 0004C
           52 DD 0004E
00000000G 00 01 FB 00050
           00000000G 00 16 00057 1$:
50          52 D0 0005D
           04 00060
           0000 00061 2$:
50          08 AC D0 00063
50          04 A0 D0 00067
           F0 A0 9F 0006B
           F4 A0 9F 0006E
           F8 A0 9F 00071
           FC A0 9F 00074
           04 DD 00077
           5E DD 00079
00000000G 7E 04 AC 7D 0007B
00          03 FB 0007F
           04 00086

```

.TITLE FOR\$IO\_END FORTTRAN End I/O statement  
.IDENT \1-011\

.EXTRN FOR\$\$A\_CUR\_LUB, FOR\$\$AA\_UDF\_PR9  
.EXTRN FOR\$\$ERR\_ENDHND  
.EXTRN FOR\$\$CB\_POP, FOR\$\$SIGNAL

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

.ENTRY FOR\$IO\_END, Save R2, R3, R4, R5, R11

```

MOVAB FOR$$AA_UDF_PR9, R3 : 0149
SUBL2 #12, SP :
CLRL L_INCR_DEPTH : 0200
CLRQ A_END_ADR :
CLRL L_UNWIND_ACTION :
MOVAL 2$, (FP) :
MOVL FOR$$A_CUR_LUB, CCB : 0219
CLRL L_UNWIND_ACTION : 0226
MOVL -T40(CCB), A_ERR_ADR : 0227
MOVL -136(CCB), A_END_ADR : 0228
CLRL L_INCR_DEPTH : 0229
MOVL -T80(CCB), USER_FRAME : 0234
MOVL -188(CCB), (USER_FRAME) : 0235
MOVZBL -143(CCB), R0 : 0243
MOVL FOR$$AA_UDF_PR9[R0], R0 : 0242
JSB FOR$$AA_UDF_PR9[R0] :
MOVZBL -144(CCB), IOSTAT : 0251
BEQL 1$ : 0253
PUSHL IOSTAT :
CALLS #1, FOR$$SIGNAL :
JSB FOR$$CB_POP : 0261
MOVL IOSTAT, R0 : 0262
RET : 0263
.WORD Save nothing : 0200
MOVL 8(AP), R0 :
MOVL 4(R0), R0 :
PUSHAB L_INCR_DEPTH :
PUSHAB A_END_ADR :
PUSHAB A_ERR_ADR :
PUSHAB L_UNWIND_ACTION :
PUSHL #4 :
PUSHL SP :
MOVQ 4(AP), -(SP) :
CALLS #3, FOR$$ERR_ENDHND :
RET :

```

: Routine Size: 135 bytes,    Routine Base: \_FOR\$CODE + 0000

```

: 201            0264 1
: 202            0265 1 END
: 203            0266 1
: 204            0267 0 ELUDOM

```

!End of module FOR\$IO\_END

## PSECT SUMMARY

```
:  
: Name Bytes Attributes  
: _FOR$CODE 135 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
```

## Library Statistics

```
:  
: File Total Symbols Loaded Percent Pages Mapped Processing Time  
: $255$DUA28:[SYSLIB]STARLET.L32;1 9776 1 0 581 00:01.0  
: -$255$DUA28:[FORRTL.OBJ]FORLIB.L32;1 711 182 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$:FORIOEND/OBJ=OBJ$:FORIOEND MSRC$:FORIOEND/UPDATE=(ENH$:FORIOEND)
```

```
: Size: 135 code + 0 data bytes  
: Run Time: 00:05.4  
: Elapsed Time: 00:20.9  
: Lines/CPU Min: 2961  
: Lexemes/CPU-Min: 6776  
: Memory Used: 79 pages  
: Compilation Complete
```



FORINTLND LIS

FORMSG LIS

FORIOBEG LIS

FORTOEND LIS

FORLEX LIS

FORMLTAB LIS

FORINQUIR LIS

FORIOELEM LIS

FORDATE LIS

FORLIB LIS