


```

FFFFFFFFF      IIIIII      000000      HH      HH      NN      NN      DDDDDDDD      LL      RRRRRRRR
FFFFFFFFF      IIIIII      000000      HH      HH      NN      NN      DDDDDDDD      LL      RRRRRRRR
FF           II           00           00      HH      HH      NN      NN      DD           DD      LL      RR      RR
FF           II           00           00      HH      HH      NN      NN      DD           DD      LL      RR      RR
FF           II           00           00      HH      HH      NN      NN      DD           DD      LL      RR      RR
FF           II           00           00      HH      HH      NN      NN      DD           DD      LL      RR      RR
FFFFFFFFF      II           00           00      HHHHHHHHHH      NN      NN      NN      DD           DD      LL      RRRRRRRR
FFFFFFFFF      II           00           00      HHHHHHHHHH      NN      NN      NN      DD           DD      LL      RRRRRRRR
FF           II           00           00      HH           HH      NN      NN      DD           DD      LL      RR      RR
FF           II           00           00      HH           HH      NN      NN      DD           DD      LL      RR      RR
FF           II           00           00      HH           HH      NN      NN      DD           DD      LL      RR      RR
FF           II           00           00      HH           HH      NN      NN      DD           DD      LL      RR      RR
FF           IIIIII      000000      HH           HH      NN      NN      DDDDDDDD      LLLLLLLLLL      RR      RR
FF           IIIIII      000000      HH           HH      NN      NN      DDDDDDDD      LLLLLLLLLL      RR      RR

```

```

LL           IIIIII      SSSSSSSS
LL           IIIIII      SSSSSSSS
LL           II           SS
LL           II           SS
LL           II           SS
LL           II           SS
LL           II           SSSSSS
LL           II           SSSSSS
LL           II           SS
LL           II           SS
LL           II           SS
LL           II           SS
LLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLL      IIIIII      SSSSSSSS

```

```

1 0001 0 %TITLE 'EDT$FIOHNDLR - file I/O condition handler'
2 0002 0 MODULE EDT$FIOHNDLR ( ! file I/O condition hanndler
3 0003 0 IDENT = 'V04-000' ! File: FIOHNDLR.B32 Edit: JBS1016
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1987, 1984 BY *
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 1 * ALL RIGHTS RESERVED. *
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 1 * TRANSFERRED. *
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 1 * CORPORATION. *
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 **
32 0032 1 FACILITY: EDT -- The DEC Standard Editor
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 Condition handler enabled for all file I/O
37 0037 1
38 0038 1 ENVIRONMENT: Runs on Vax/Vms
39 0039 1
40 0040 1 AUTHOR: Shelly T. Solomon, CREATION DATE: 04-Jan-1982
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SIS 04-Jan-1982
45 0045 1 1-002 - Fixed command file error. STS 14-Jan-1982
46 0046 1 1-003 - Catch open errors. STS 25-Feb-1982
47 0047 1 1-004 - Add literals for callablre parameters. STS 08-Mar-1982
48 0048 1 1-005 - Change EDT$$G_CMD to EDT$$V_OPTIONS. JBS 18-Mar-1982
49 0049 1 1-006 - If we unwind, make sure there is a bad status code in R0. JBS 26-mar-1982
50 0050 1 1-007 - Revise offsets into the signal args to reflect the new signaling
51 0051 1 in FILEIO -- the RMS STS is now always in the fourth longword. JBS 29-Mar-1982
52 0052 1 1-008 - Put in lots more error checking. JBS 29-Mar-1982
53 0053 1 1-009 - Catch read and write errors and unwind. STS 14-Jul-1982
54 0054 1 1-010 - Set abort write flag if error on get. STS 05-Aug-1982
55 0055 1 1-011 - Make use of bit masks for options word. STS 17-Aug-1982
56 0056 1 1-012 - Fix up references to EDT$M_ symbols. JBS 23-Aug-1982
57 0057 1 1-013 - Don't alter returned R0 when unwinding. JBS 23-Feb-1983

```

EDTSFIOHDLR
V04-000

EDTSFIOHDLR - file I/O condition handler

C 1
16-Sep-1984 00:25:45
14-Sep-1984 12:23:08

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]FIOHDLR.B32;1 Page 2 (1)

```
: 58      0058 1 : 1-014 - If we resignal, arrange to rebuild the screen if we are continued. JBS 22-Mar-1983
: 59      0059 1 : 1-015 - Remove some unreachable code. JBS 23-Mar-1983
: 60      0060 1 : 1-016 - Improve the appearance of the listing. JBS 14-Jun-1983
: 61      0061 1 : --
: 62      0062 1
```

EDT
V04

: R

:
:

```
64 0063 1 %SBTTL 'Declarations'  
65 0064 1  
66 0065 1 : TABLE OF CONTENTS:  
67 0066 1 :  
68 0067 1  
69 0068 1 REQUIRE 'EDT$SRC:TRAROUNAM';  
70 0507 1  
71 0508 1 FORWARD ROUTINE  
72 0509 1 EDT$FIOHDLR;  
73 0510 1  
74 0511 1 :  
75 0512 1 : INCLUDE FILES:  
76 0513 1 :  
77 0514 1  
78 0515 1 REQUIRE 'EDT$SRC:EDTREQ';  
79 0650 1  
80 0651 1 REQUIRE 'EDT$SRC:SYSSYM';  
81 0681 1  
82 0682 1 :  
83 0683 1 : MACROS:  
84 0684 1 :  
85 0685 1 : NONE  
86 0686 1 :  
87 0687 1 : EQUATED SYMBOLS:  
88 0688 1 :  
89 0689 1 : NONE  
90 0690 1 :  
91 0691 1 : FIELDS:  
92 0692 1 :  
93 0693 1 : NONE  
94 0694 1 :  
95 0695 1 : OWN STORAGE:  
96 0696 1 :  
97 0697 1 : in the routine  
98 0698 1 :  
99 0699 1 : EXTERNAL REFERENCES:  
100 0700 1 :  
101 0701 1 : in the routine  
102 0702 1 :
```

```

: 104 0703 1 %SBTTL 'EDT$$FIOHNDLR - condition handler for file I/O'
: 105 0704 1
: 106 0705 1 GLOBAL ROUTINE EDT$$FIOHNDLR (           ! Condition handler for file I/O
: 107 0706 1     SIG,                               ! Signal arguments
: 108 0707 1     MECH,                               ! Mechanism arguments
: 109 0708 1     ENBL                               ! Enable arguments
: 110 0709 1     ) =
: 111 0710 1
: 112 0711 1 ++
: 113 0712 1  FUNCTIONAL DESCRIPTION:
: 114 0713 1
: 115 0714 1     This is the condition handler for file I/O.  It intercepts
: 116 0715 1     the non-fatal conditions, such as file not found, and unwinds
: 117 0716 1     to the caller of EDT$$CALLFIO.  Errors not recognized are
: 118 0717 1     resigaled to the caller to EDT$EDIT.  Just in case that caller
: 119 0718 1     has a handler which does a continue, arrange to reset and rebuild
: 120 0719 1     the screen if the screen updater regains control, and stop the
: 121 0720 1     working message so it won't interfere with whatever the user
: 122 0721 1     may have in mind.
: 123 0722 1
: 124 0723 1  FORMAL PARAMETERS:
: 125 0724 1
: 126 0725 1     Usual for condition handlers.
: 127 0726 1
: 128 0727 1  IMPLICIT INPUTS:
: 129 0728 1
: 130 0729 1     NONE
: 131 0730 1
: 132 0731 1  IMPLICIT OUTPUTS:
: 133 0732 1
: 134 0733 1     EDT$$G_IOFI_NFND           1 = error was "file not found"
: 135 0734 1     EDT$$G_ABT_WRITE        1 = abort the current WRITE or EXIT operation.
: 136 0735 1     EDT$$G_SCR_CHGD         2 = reset and rebuild screen on continue
: 137 0736 1
: 138 0737 1  COMPLETION STATUS:
: 139 0738 1
: 140 0739 1     As usual for condition handlers
: 141 0740 1
: 142 0741 1  SIDE EFFECTS:
: 143 0742 1
: 144 0743 1     NONE
: 145 0744 1
: 146 0745 1  --
: 147 0746 1
: 148 0747 2  BEGIN
: 149 0748 2
: 150 0749 2  EXTERNAL ROUTINE
: 151 0750 2     EDT$$STOP_WKINGMSG : NOVALUE;           ! Stop the WORKING message
: 152 0751 2
: 153 0752 2  EXTERNAL
: 154 0753 2     EDT$$G_ABT_WRITE,
: 155 0754 2     EDT$$G_IO_FNAM,
: 156 0755 2     EDT$$G_IO_STS,
: 157 0756 2     EDT$$G_IO_STV,
: 158 0757 2     EDT$$G_IOFI_NFND,
: 159 0758 2     EDT$$G_SCR_REBUILD;
: 160 0759 2
```

```
161 0760 2      EXTERNAL LITERAL
162 0761 2      EDT$K_COMMAND_FILE,
163 0762 2      EDT$K_INPUT_FILE,
164 0763 2      EDT$K_INCLUDE_FILE,
165 0764 2      EDT$K_WRITE_FILE,
166 0765 2      EDT$K_OUTPUT_FILE,
167 0766 2      EDT$K_PUT,
168 0767 2      EDT$K_GET,
169 0768 2      EDT$K_OPEN_INPUT,
170 0769 2      EDT$K_OPEN_OUTPUT_NOSEQ,
171 0770 2      EDT$K_OPEN_OUTPUT_SEQ;
172 0771 2
173 0772 2      MAP
174 0773 2      SIG : REF VECTOR,
175 0774 2      MECH : REF VECTOR,
176 0775 2      ENBL : REF VECTOR;
177 0776 2
178 0777 2      BIND
179 0778 2      FCODE = .ENBL [2],
180 0779 2      RET_VALUE = MECH [3],
181 0780 2      STREAM = .ENBL [1];
182 0781 2
183 0782 2      LOCAL
184 0783 2      STATUS;
185 0784 2
186 0785 3      IF ((.SIG [0] EQL 1) AND (.SIG [1] EQL S$$_UNWIND))
187 0786 3      THEN
188 0787 3      BEGIN
189 0788 3      +
190 0789 3      | We are unwinding.
191 0790 3      -
192 0791 3      |   RETURN (S$$_CONTINUE);
193 0792 3      |   END;
194 0793 3
195 0794 3      IF ((.SIG [0] NEQ 7) OR (.SIG [2] NEQ 1))
196 0795 3      THEN
197 0796 3      BEGIN
198 0797 3      +
199 0798 3      | If we are continued, reset and rebuild the screen. There is no telling
200 0799 3      | what a user might do in his own handler before continuing us.
201 0800 3      -
202 0801 3      |   EDT$$G_SCR_REBUILD = 2;
203 0802 3      +
204 0803 3      | Also, stop the working message, so it won't interfere with whatever the
205 0804 3      | user may have in mind.
206 0805 3      -
207 0806 3      |   EDT$$STOP_WKINGMSG ();
208 0807 3      |   RETURN (S$$_RESIGNAL);
209 0808 3      |   END;
210 0809 2
211 0810 2      +
212 0811 2      | This is probably a signal from the file I/O routine.
213 0812 2      | Set up the message vector for the call to EDT$$IO_ERRMSG
214 0813 2      | and save the filename signalled with
215 0814 2      -
216 0815 2      |   EDT$$G_IO_STS = .SIG [4];
217 0816 2      |   EDT$$G_IO_STV = .SIG [5];
```

```
218 0817 2 EDT$$A_IO_FNAM = .SIG [3];
219 0818 2
220 0819 2 SELECTONE .STREAM OF
221 0820 2 SET
222 0821 2
223 0822 2 [EDT$K_COMMAND_FILE] :
224 0823 3 BEGIN
225 0824 3
226 0825 4 IF (.SIG [4] EQL RMS$_FNF) ! File not found
227 0826 3 THEN
228 0827 4 BEGIN
229 0828 4 EDT$$G_IOFI_NFND = 1; ! set file not found flag
230 0829 4 SETUNWIND (?); ! everything O.K., continue (caller will print message)
231 0830 4 RETURN (SS$_CONTINUE);
232 0831 4 END
233 0832 3 ELSE
234 0833 3 !+ Not "file not found", resignal the error.
235 0834 3 -
236 0835 3 BEGIN
237 0836 4
238 0837 4 !+
239 0838 4 If we are continued, reset and rebuild the screen. There is no telling
240 0839 4 what a user might do in his own handler before continuing us.
241 0840 4 -
242 0841 4 EDT$$G_SCR_REBUILD = 2;
243 0842 4 !+
244 0843 4 Also, stop the working message, so it won't interfere with whatever the
245 0844 4 user may have in mind.
246 0845 4 -
247 0846 4 EDT$$STOP_WKINGMSG ();
248 0847 4 RETURN (SS$_RESIGNAL);
249 0848 3 END;
250 0849 3
251 0850 2 FND;
252 0851 2
253 0852 2 [EDT$K_INPUT_FILE] :
254 0853 3 BEGIN
255 0854 3
256 0855 4 IF (.SIG [4] EQL RMS$_FNF)
257 0856 3 THEN
258 0857 4 BEGIN
259 0858 4 EDT$$G_IOFI_NFND = 1; ! set file not found flag
260 0859 4 SETUNWIND (?); ! everything O.K., continue
261 0860 4 RETURN (SS$_CONTINUE);
262 0861 4 END
263 0862 3 ELSE
264 0863 3
265 0864 4 IF (.FCODE EQL EDT$K_GET)
266 0865 3 THEN
267 0866 4 BEGIN
268 0867 4 EDT$$G_ABT_WRITE = 1;
269 0868 4 SETUNWIND (?);
270 0869 4 RETURN (SS$_CONTINUE);
271 0870 4 END
272 0871 3 ELSE
273 0872 4 BEGIN
274 0873 4 !+
```



```
275 0874 4 | If we are continued, reset and rebuild the screen. There is no telling
276 0875 4 | what a user might do in his own handler before continuing us.
277 0876 4 | -
278 0877 4 |         EDT$$G_SCR_REBUILD = 2;
279 0878 4 | +
280 0879 4 | Also, stop the working message, so it won't interfere with whatever the
281 0880 4 | user may have in mind.
282 0881 4 | -
283 0882 4 |         EDT$$STOP_WKINGMSG ();
284 0883 4 |         RETURN (SS$_RESIGNAL);
285 0884 3 |         END;
286 0885 3 |
287 0886 2 |     END;
288 0887 2 |
289 0888 2 | [EDT$K_INCLUDE_FILE, EDT$K_WRITE_FILE, EDT$K_OUTPUT_FILE] :
290 0889 2 | BEGIN
291 0890 3 | +
292 0891 3 | Any error opening any of the above files is not considered fatal.
293 0892 3 | -
294 0893 3 |
295 0894 4 |     IF ((.FCODE EQL EDT$K_OPEN_INPUT) OR          !
296 0895 4 |         (.FCODE EQL EDT$K_OPEN_OUTPUT_NOSEQ) OR  !
297 0896 4 |         (.FCODE EQL EDT$K_OPEN_OUTPUT_SEQ))
298 0897 3 |     THEN
299 0898 4 |         BEGIN
300 0899 4 | +
301 0900 4 | If the error is "file not found" (only possible with INCLUDE) then
302 0901 4 | give a more-specific message.
303 0902 4 | -
304 0903 4 |
305 0904 4 |         IF (.SIG [4] EQL RMS$_FNF) THEN EDT$$G_IOFI_NFND = 1;
306 0905 4 |
307 0906 4 |         SETUNWIND ();
308 0907 4 |         RETURN (SS$_CONTINUE);
309 0908 4 |         END
310 0909 3 |     ELSE
311 0910 3 |
312 0911 4 |         IF ((.FCODE EQL EDT$K_PUT) OR (.FCODE EQL EDT$K_GET))
313 0912 3 |         THEN
314 0913 4 |             BEGIN
315 0914 4 |                 EDT$$G_ABT_WRITE = 1;
316 0915 4 |                 SETUNWIND ();
317 0916 4 |                 RETURN (SS$_CONTINUE);
318 0917 4 |                 END
319 0918 3 |             ELSE
320 0919 4 |                 BEGIN
321 0920 4 | +
322 0921 4 | If we are continued, reset and rebuild the screen. There is no telling
323 0922 4 | what a user might do in his own handler before continuing us.
324 0923 4 | -
325 0924 4 |                 EDT$$G_SCR_REBUILD = 2;
326 0925 4 | +
327 0926 4 | Also, stop the working message, so it won't interfere with whatever the
328 0927 4 | user may have in mind.
329 0928 4 | -
330 0929 4 |                 EDT$$STOP_WKINGMSG ();
331 0930 4 |                 RETURN (SS$_RESIGNAL);
```

```

332 0931 3      END;
333 0932
334 0933      END;
335 0934
336 0935      [OTHERWISE] :
337 0936      BEGIN
338 0937
339 0938      + If we are continued, reset and rebuild the screen. There is no telling
340 0939      - what a user might do in his own handler before continuing us.
341 0940
342 0941      EDT$$G_SCR_REBUILD = 2;
343 0942
344 0943      + Also, stop the working message, so it won't interfere with whatever the
345 0944      - user may have in mind.
346 0945
347 0946      EDT$$STOP WKINGMSG ();
348 0947      RETURN (SS$_RESIGNAL);
349 0948      END;
350 0949      TES;
351 0950
352 0951 1      END;

```

! of routine EDT\$\$FIOHNDLR

```

.TITLE EDT$FIOHNDLR EDT$FIOHNDLR - file I/O condition handler
.IDENT \V04-000\

.EXTRN EDT$$STOP WKINGMSG
.EXTRN EDT$$G_ABT_WRITE
.EXTRN EDT$$A_IO_FNAM, EDT$$G_IO_STS
.EXTRN EDT$$G_IO_STV, EDT$$G_IOFI_NFND
.EXTRN EDT$$G_SCR_REBUILD
.EXTRN EDT$K_COMMAND_FILE
.EXTRN EDT$K_INPUT_FILE
.EXTRN EDT$K_INCLUDE_FILE
.EXTRN EDT$K_WRITE_FILE
.EXTRN EDT$K_OUTPUT_FILE
.EXTRN EDT$K_PUT, EDT$K_GET
.EXTRN EDT$K_OPEN_INPUT
.EXTRN EDT$K_OPEN_OUTPUT_NOSEQ
.EXTRN EDT$K_OPEN_OUTPUT_SEQ

```

```

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

```

```

.ENTRY EDT$$FIOHNDLR, Save R2,R3,R4      : 0705
MOVL #EDT$K_GET, R4                      :
MOVL ENBL, R3                             : 0778
MOVL SIG, R2                              : 0785
CMPL (R2), #1
BNEQ 1$
CMPL 4(R2), #2336
BNEQ 1$
BRW 13$
CMPL (R2), #7                             : 0794
BNEQ 6$
CMPL 8(R2), #1
BNEQ 6$

```

```

001C 00000
54 00000000G 8F D0 00002
53 0C AC D0 00009
52 04 AC D0 0000D
01 62 D1 00011
0D 12 00014
0000920 8F 04 A2 D1 00016
03 12 0001E
00C2 31 00020
07 62 D1 00023 1$:
72 12 00026
01 08 A2 D1 00028
6C 12 0002C

```

00000000G	00	10	A2	D0	0002E	MOVL	16(R2), EDT\$\$G_IO_STS	:	0815
00000000G	00	14	A2	D0	00036	MOVL	20(R2), EDT\$\$G_IO_STV	:	0816
00000000G	0C	0C	A2	D0	0003E	MOVL	12(R2), EDT\$\$A_IO_FNAM	:	0817
	50	04	B3	D0	00046	MOVL	@4(R3), R0	:	0819
00000000G	8F		50	D1	0004A	CMPL	R0, #EDTSK_COMMAND_FILE	:	0822
			0C	12	00051	BNEQ	2\$:	
00018292	8F	10	A2	D1	00053	CMPL	16(R2), #98962	:	0825
			76	12	0005B	BNEQ	10\$:	
			13	11	0005D	BRB	3\$:	0828
00000000G	8F		50	D1	0005F	CMPL	R0, #EDTSK_INPUT_FILE	:	0852
			19	12	00066	BNEQ	5\$:	
00018292	8F	10	A2	D1	00068	CMPL	16(R2), #98962	:	0855
			09	12	00070	BNEQ	4\$:	
00000000G	00		01	D0	00072	MOVL	#1, EDT\$\$G_IOFI_NFND	:	0858
			61	11	00079	BRB	12\$:	0859
	54	08	B3	D1	0007B	CMPL	@8(R3), R4	:	0864
			52	11	0007F	BRB	10\$:	
00000000G	8F		50	D1	00081	CMPL	R0, #EDTSK_INCLUDE_FILE	:	0888
			12	13	00088	BEQL	7\$:	
00000000G	8F		50	D1	0008A	CMPL	R0, #EDTSK_WRITE_FILE	:	
			09	13	00091	BEQL	7\$:	
00000000G	8F		50	D1	00093	CMPL	R0, #EDTSK_OUTPUT_FILE	:	
			4D	12	0009A	BNEQ	14\$:	
	50	08	B3	D0	0009C	MOVL	@8(R3), R0	:	0894
00000000G	8F		50	D1	000A0	CMPL	R0, #EDTSK_OPEN_INPUT	:	
			12	13	00CA7	BEQL	8\$:	
00000000G	8F		50	D1	000A9	CMPL	R0, #EDTSK_OPEN_OUTPUT_NOSEQ	:	0895
			09	13	000B0	BEQL	8\$:	
00000000G	8F		50	D1	000B2	CMPL	R0, #EDTSK_OPEN_OUTPUT_SEQ	:	0896
			0C	12	000B9	BNEQ	9\$:	
00018292	8F	10	A2	D1	000BB	CMPL	16(R2), #98962	:	0904
			17	12	000C3	BNEQ	12\$:	
			AB	11	000C5	BRB	3\$:	
00000000G	8F		50	D1	000C7	CMPL	R0, #EDTSK_PUT	:	0911
			05	13	000CE	BEQL	11\$:	
	54		50	D1	000D0	CMPL	R0, R4	:	
			14	12	000D3	BNEQ	14\$:	
00000000G	00		01	D0	000D5	MOVL	#1, EDT\$\$G_ABT_WRITE	:	0914
			7E	7C	000DC	CLRQ	-(SP)	:	0915
00000000G	00		02	FB	000DE	CALLS	#2, SYSSUNWIND	:	
	50		01	D0	000E5	MOVL	#1, R0	:	0916
			04	000E8	RET			:	
00000000G	00		02	D0	000E9	MOVL	#2, EDT\$\$G_SCR_REBUILD	:	0941
00000000G	00		00	FB	000F0	CALLS	#0, EDT\$\$STOP_WKINGMSG	:	0946
	50	0918	8F	3C	000F7	MOVZWL	#2328, R0	:	0947
			04	000FC	RET			:	0951

; Routine Size: 253 bytes, Routine Base: _EDT\$CODE + 0000

: 353 0952 1
: 354 0953 1 !<BLF/PAGE>

: 356 0954 1 END ! of module EDT\$FIOHNDLR
 : 357 0955 1
 : 358 0956 0 ELUDOM

.EXTRN SYSSUNWIND

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	253	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	0	0	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	4	0	581	00:04.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:FIOHNDLR/OBJ=OBJ\$:FIOHNDLR MSRC\$:FIOHNDLR.B32/UPDATE=(ENHS:FIOHNDLR)

: Size: 253 code + 0 data bytes
 : Run Time: 00:23.1
 : Elapsed Time: 00:30.9
 : Lines/CPU Min: 2486
 : Lexemes/CPU-Min: 7427
 : Memory Used: 123 pages
 : Compilation Complete

0134 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

