



```

LL      IIIIII  NN      NN  CCCCCCCC  LL
LL      IIIIII  NN      NN  CCCCCCCC  LL
LL      II      NN      NN  CC        LL
LL      II      NN      NN  CC        LL
LL      II      NNNN     NN  CC        LL
LL      II      NNNN     NN  CC        LL
LL      II      NN  NN    NN  CC        LL
LL      II      NN  NN    NN  CC        LL
LL      II      NN      NNNN  CC        LL
LL      II      NN      NNNN  CC        LL
LL      II      NN      NN    CC        LL
LL      II      NN      NN    CC        LL
LLLLLLLLLLLL  IIIIII  NN      NN  CCCCCCCC  LLLLLLLLLL
LLLLLLLLLLLL  IIIIII  NN      NN  CCCCCCCC  LLLLLLLLLL

```

```

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
LLLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIII  SSSSSSSS

```



```

1 0001 0 %TITLE 'EDT$LINCL - INCLUDE line-mode command'
2 0002 0 MODULE EDT$LINCL ( ! Line-mode INCLUDE command
3 0003 0 IDENT = 'V04-000' ! File: LINCL.BLI Edit: JBS1028
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 1 * ALL RIGHTS RESERVED. *
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 1 * TRANSFERRED. *
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 1 * CORPORATION. *
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: EDT -- The DEC Standard Editor
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module executes the line mode INCLUDE command.
37 0037 1
38 0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 30-JAN-1981. This module was created by
45 0045 1 extracting the routine EDT$$INCL CMD from the module EXEC.BLI.
46 0046 1 1-002 - Change from IO STS and IO_FNF to EDT$$G_IOFI_NFND .
47 0047 1 JBS 19-Feb-1981
48 0048 1 1-003 - Fix module name. JBS 20-Mar-1981
49 0049 1 1-004 - Use new message codes. JBS 04-Aug-1981
50 0050 1 1-005 - Change I/O calls to use EDT$FILEIO. STS 26-Dec-1981
51 0051 1 1-006 - Change call to fileio to callfio. STS 06-Jan-1982
52 0052 1 1-007 - Fix DSCSA_POINTER macro. STS 14-Jan-1982
53 0053 1 1-008 - Add rhb descr. definitions. STS 22-Jan-1982
54 0054 1 1-009 - Add a return value for line insertion success. SMB 3-Feb-1982
55 0055 1 1-010 - Deallocate dynamic descriptors. STS 10-Feb-1982
56 0056 1 1-011 - Pass filename to edt$$fiopn_err. STS 25-Feb-1982
57 0057 1 1-012 - Add literals for callable parameters. STS 08-Mar-1982

```

```
58 0058 1 1-013 - File I/O routines return a status. JBS 26-Mar-1982
59 0059 1 1-014 - Correct a typo in edit 1-013. JBS 29-Mar-1982
60 0060 1 1-015 - Print a message on a CLOSE error. JBS 12-Apr-1982
61 0061 1 1-016 - Change PDP-11 filenames to uppercase. SMB 19-Apr-1982
62 0062 1 1-017 - Add error message for no filename. SMB 21-Apr-1982
63 0063 1 1-018 - Check for record too long. JBS 02-Jun-1982
64 0064 1 1-019 - Pass default name in RHB parameter. JBS 15-Jun-1982
65 0065 1 1-020 - Add control C checking. SMB 30-Jun-1982
66 0066 1 1-021 - Stop processing on bad select range. SMB 01-Jul-1982
67 0067 1 1-022 - Set EDT$$G_SCR_CHGD if the screen is changed. JBS 07-Jul-1982
68 0068 1 1-023 - Change names of include file messages. SMB 13-Jul-1982
69 0069 1 1-024 - Print out error messages on GETs. STS 20-Jul-1982
70 0070 1 1-025 - Don't check RMS status on 11's since error message has
71 0071 1 already been printed. STS 22-Jul-1982
72 0072 1 1-026 - Don't set EDT$$G_SCR_CHGD, new screen logic doesn't need it. JBS 09-Oct-1982
73 0073 1 1-027 - Put edt$$rng_pos1rst in line. STS 11-Oct-1982
74 0074 1 1-028 - Make sure select is at the beginning of the line. JBS 12-Jul-1983
75 0075 1 --
76 0076 1
```

```
78 0077 1 %SBTTL 'Declarations'  
79 0078 1  
80 0079 1 : TABLE OF CONTENTS:  
81 0080 1 :  
82 0081 1 :  
83 0082 1 REQUIRE 'EDTSRC:TRAROUNAM';  
84 0521 1  
85 0522 1 FORWARD ROUTINE  
86 0523 1 EDT$INCL_CMD : NOVALUE;  
87 0524 1  
88 0525 1 :  
89 0526 1 : INCLUDE FILES:  
90 0527 1 :  
91 0528 1 :  
92 0529 1 REQUIRE 'EDTSRC:EDTREQ';  
93 0664 1  
94 0665 1 :+  
95 0666 1 : VMS system symbols  
96 0667 1 :-  
97 0668 1  
98 L 0669 1 %IF %BLISS (BLISS32)  
99 0670 1 %THEN  
100 0671 1  
101 0672 1 REQUIRE 'EDTSRC:SYSSYM';  
102 0702 1  
103 0703 1 %FI  
104 0704 1  
105 0705 1 :  
106 0706 1 : MACROS:  
107 0707 1 :  
108 0708 1 :  
109 0709 1 : EQUATED SYMBOLS:  
110 0710 1 :  
111 0711 1 : NONE  
112 0712 1 :  
113 0713 1 : LITERALS  
114 0714 1 :  
115 0715 1 :  
116 0716 1 : EXTERNAL LITERAL  
117 0717 1 : EDT$K_OPEN_INPUT,  
118 0718 1 : EDT$K_CLOSE,  
119 0719 1 : EDT$K_GET,  
120 0720 1 : EDT$K_INCLUDE_FILE;  
121 0721 1 :  
122 0722 1 :  
123 0723 1 : OWN STORAGE:  
124 0724 1 :  
125 0725 1 : NONE  
126 0726 1 :  
127 0727 1 : EXTERNAL REFERENCES:  
128 0728 1 :  
129 0729 1 : In the routine
```

! Process the INCLUDE command

```

131 0730 1 %SBTTL 'EDT$$INCL_CMD - Line mode INCLUDE command'
132 0731 1
133 0732 1 GLOBAL ROUTINE EDT$$INCL_CMD          ! Line mode INCLUDE command
134 0733 1   : NOVALUE =
135 0734 1
136 0735 1 !++
137 0736 1 ! FUNCTIONAL DESCRIPTION:
138 0737 1
139 0738 1     Command processing routing for the INCLUDE command. Attempt to open the file.
140 0739 1     If it succeeds then position as indicated by the range and read the file
141 0740 1     into the buffer in front of that line.
142 0741 1
143 0742 1 ! FORMAL PARAMETERS:
144 0743 1
145 0744 1     NONE
146 0745 1
147 0746 1 ! IMPLICIT INPUTS:
148 0747 1
149 0748 1     EDT$$LN_BUF
150 0749 1     EDT$$EXE_CURCMD
151 0750 1
152 0751 1 ! IMPLICIT OUTPUTS:
153 0752 1
154 0753 1     EDT$$SEL_POS
155 0754 1     EDT$$TXT_ONSCR
156 0755 1
157 0756 1 ! ROUTINE VALUE:
158 0757 1
159 0758 1     NONE
160 0759 1
161 0760 1 ! SIDE EFFECTS:
162 0761 1
163 0762 1     Reads a text file into the work file system, or prints an error message.
164 0763 1
165 0764 1 !--
166 0765 1
167 0766 2     BEGIN
168 0767 2
169 0768 2     EXTERNAL ROUTINE
170 0769 2     EDT$$STOP_WKINGMSG,
171 0770 2     EDT$$CHK_CC,
172 0771 2     EDT$$FMT_MSG,
173 0772 2     EDT$$CALCFIO,
174 0773 2     EDT$$RNG_REPOS,
175 0774 2     EDT$$END_INS,
176 0775 2     EDT$$INS_LN,
177 0776 2     EDT$$START_INS,
178 0777 2     EDT$$FIOPN_ERR;
179 0778 2
180 L 0779 2 %IF %BLISS (BLISS32)
181 0780 2 %THEN
182 0781 2
183 0782 2     EXTERNAL ROUTINE
184 0783 2     STR$FREE1_DX;
185 0784 2
186 0785 2 %FI
187 0786 2

```

```

188 0787 2 EXTERNAL
189 0788 2 EDT$$G_RNG_FRSTLN,
190 0789 2 EDT$$A_CUR_BUF : REF TBCB_BLOCK,
191 0790 2 EDT$$Z_RNG_ORIGPOS : POS_BLOCK,
192 0791 2 EDT$$A_IO_FNAM,
193 0792 2 EDT$$G_CC_DONE,
194 0793 2 EDT$$G_IOFI_NFND,
195 0794 2 EDT$$A_EXE_CURCMD : REF NODE_BLOCK,
196 0795 2 EDT$$G_TXT_ONSCR,
197 0796 2 EDT$$T_LN_BUF,
198 0797 2 EDT$$A_SEC_POS;
199 0798
200 0799 2 MESSAGES ((INCFILNEX, ERRINPFIL, INCFILOPN, INCFILCLO, NOFILSPC, RECTOOBIG));
201 0800
202 0801 2 LOCAL
203 0802 2 FILE_DESC : BLOCK [8, BYTE],
204 0803 2 RHB_DESC : BLOCK [8, BYTE],
205 0804 2 GET,
206 0805 2 STRADR,
207 0806 2 STATUS,
208 0807 2 STRLEN,
209 0808 2 RECORD_TOO_BIG;
210 0809
211 0810 2 !+
212 0811 2 ! If this is VMS, then we are going to pass a descriptor to the
213 0812 2 ! file i/o routine, so set up the type and class
214 0813 2 !-
215 0814 2
216 L 0815 2 %IF %BLISS (BLISS32)
217 0816 2 %THEN
218 0817 2 RHB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
219 0818 2 RHB_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
220 0819 2 FILE_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
221 0820 2 FILE_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
222 0821 2 %FI
223 0822 2
224 0823 2 RECORD_TOO_BIG = 0;
225 0824 2 !+
226 0825 2 ! Open the file.
227 0826 2 !-
228 0827 2 RHB_DESC [DSC$A_POINTER] = 0;
229 0828 2 RHB_DESC [DSC$W_LENGTH] = 0;
230 0829 2 FILE_DESC [DSC$A_POINTER] = 0;
231 0830 2 FILE_DESC [DSC$W_LENGTH] = 0;
232 0831 2 STRLEN = .EDT$$A_EXE_CURCMD [FSPCLN];
233 0832 2 STRADR = .EDT$$A_EXE_CURCMD [FILSPEC];
234 0833 2 !+
235 0834 2 ! File specification must be given for INCLUDE
236 0835 2 !-
237 0836 2
238 0837 2 IF (.STRLEN EQL 0)
239 0838 2 THEN
240 0839 2 BEGIN
241 0840 2 EDT$$FMT_MSG (EDT$_NOFILSPC);
242 0841 2 RETURN (0);
243 0842 2 END;
244 0843 2

```

```

! CNTRL/C aborted the command
! File not found flag
! Pointer to the current command.
! Force 'press return to continue' if under EXT
! Line buffer
! Select position in line buffer

```

```

! length of filename
! Address of filename string

```

```
245 0844 2 STRING_DESC (FILE_DESC, STRLEN, .STRADR);
246 0845 2 STATUS = EDT$$CALLFIO (EDT$K_OPEN_INPJT, EDT$K_INCLUDE_FILE, FILE_DESC, RMB_DESC);
247 0846 2
248 0847 2 IF .STATUS
249 0848 2 THEN
250 0849 2 BEGIN
251 0850 2 !+
252 0851 2 ! File was opened. Position to start of the range.
253 0852 2 !-
254 0853 2 EDT$$G_RNG_FRSTLN = 1;
255 0854 2 EDT$$CPY_MEM (POS_SIZE, .EDT$$A_CUR_BUF, EDT$$Z_RNG_ORIGPOS);
256 0855 2
257 0856 2 IF ( NOT EDT$$RNG_REPOS (.EDT$$A_EXE_CURCMD [RANGE1])) THEN RETURN (0);
258 0857 2
259 0858 2 !+
260 0859 2 ! Insert all the lines from the file at the current position.
261 0860 2 !-
262 0861 2 EDT$$START_INS ();
263 0862 2 STATUS = 1;
264 0863 2
265 0864 2 WHILE (.STATUS NEQ 0) DO
266 0865 2 BEGIN
267 0866 2 GET = EDT$$CALLFIO (EDT$K_GET, EDT$K_INCLUDE_FILE, FILE_DESC, RMB_DESC);
268 0867 2
269 L 0868 2 %IF %BLISS (BLISS32)
270 0869 2 %THEN
271 0870 2
272 0871 2 IF (.GET EQL RMS$ EOF) THEN EXITLOOP;
273 0872 2
274 0873 2 IF ( NOT .GET) THEN EDT$$FIOPN_ERR (EDT$ _ERRINPFIL, .EDT$$A_IO_FNAM);
275 0874 2
276 U 0875 2 %ELSE
277 U 0876 2
278 U 0877 2 IF ( NOT .GET) THEN EXITLOOP;
279 U 0878 2
280 0879 2 %F!
281 0880 2
282 0881 2 IF (EDT$$CHK_CC ())
283 0882 2 THEN
284 0883 2 BEGIN
285 0884 2 STATUS = EDT$$CALLFIO (EDT$K_CLOSE, EDT$K_INCLUDE_FILE, 0, 0);
286 0885 2
287 0886 2 IF ( NOT .STATUS) THEN EDT$$FIOPN_ERR (EDT$ _INCFILCLO, FILE_DESC);
288 0887 2
289 0888 2 EDT$$G_CC_DONE = 1;
290 0889 2 RETURN (0);
291 0890 2 END;
292 0891 2
293 0892 2 IF (.FILE_DESC [DSC$W_LENGTH] GTR 255) THEN RECORD_TOO_BIG = 1;
294 0893 2
295 0894 2 EDT$$A_SEL_POS = EDT$$T_LN_BUF;
296 0895 2 STATUS = EDT$$INS_LN (.FILE_DESC [DSC$A_POINTER], MIN (.FILE_DESC [DSC$W_LENGTH], 255));
297 0896 2 END;
298 0897 2
299 0898 2 EDT$$END_INS ();
300 0899 2 !+
301 0900 2 ! And close the file.
```



```

302 0901 3 :-
303 0902 3 EDT$$STOP WKINGMSG ();
304 0903 3 STATUS = EDT$$CALLFIO (EDT$K_CLOSE, EDT$K_INCLUDE_FILE, 0, 0);
305 0904 3
306 0905 3 IF ( NOT .STATUS) THEN EDT$$FIOPN_ERR (EDT$_INCFILCLO, FILE_DESC);
307 0906 3
308 0907 3 END
309 0908 2 ELSE
310 0909 2 BEGIN
311 0910 2
312 0911 2 !+ The open failed, put out an error message.
313 0912 2 !-
314 0913 2 EDT$$FIOPN_ERR (
315 0914 2
316 0915 2 IF .EDT$$G_IOFI_NFND THEN EDT$_INCFILNEX ELSE EDT$_INCFILOPN, FILE_DESC);
317 0916 2
318 0917 2 END;
319 0918 2
320 0919 2 !+
321 0920 2 ! If we print the message "record too big" then force "press return to continue",
322 0921 2 ! since the message would otherwise be lost in the update of the screen that we
323 0922 2 ! must do.
324 0923 2 !-
325 0924 2
326 0925 2 IF .RECORD_TOO_BIG
327 0926 2 THEN
328 0927 2 BEGIN
329 0928 2 EDT$$FMT MSG (EDT$_RECTOOBIG);
330 0929 2 EDT$$G_TXT_ONSCR =-1;
331 0930 2 END;
332 0931 2
333 0932 2 !+
334 0933 2 ! On VMS deallocate all dynamic descriptors used
335 0934 2 !-
336 0935 2
337 L 0936 2 %IF %BLISS (BLISS32)
338 0937 2 %THEN
339 0938 2 STR$FREE1_DX (FILE_DESC);
340 0939 2 STR$FREE1_DX (RMB_DESC);
341 0940 2 %FI
342 0941 2
343 0942 1 END;

```

! of routine EDT\$\$INCL\_CMD

```

.TITLE EDT$LINCL EDT$LINCL - INCLUDE line-mode command
.IDENT \V04-000\

.EXTRN EDT$K_OPEN_INPUT
.EXTRN EDT$K_CLOSE, EDT$K_GET
.EXTRN EDT$K_INCLUDE_FILE
.EXTRN EDT$$STOP WKINGMSG
.EXTRN EDT$$CHK_CC, EDT$$FMT MSG
.EXTRN EDT$$CALLFIO, EDT$$RNG_REPOS
.EXTRN EDT$$SEND_INS, EDT$$INS_LN
.EXTRN EDT$$START_INS, EDT$$FIOPN_ERR
.EXTRN STR$FREE1_DX, EDT$$G_RNG_FRSTLN
.EXTRN EDT$$A_CUR_BUF, EDT$$Z_RNG_ORIGPOS

```

					.EXTRN	EDT\$\$A_IO_FNAM, EDT\$\$G_CC_DONE	
					.EXTRN	EDT\$\$G_IOFI_NFND	
					.EXTRN	EDT\$\$A_EXE_CURCMD	
					.EXTRN	EDT\$\$G_TXT_ONSCR	
					.EXTRN	EDT\$\$T_LN_BUF, EDT\$\$A_SEL_POS	
					.EXTRN	EDT\$-INCFILNEX, EDT\$-ERRINPFIL	
					.EXTRN	EDT\$-INCFILOPN, EDT\$-INCFILCLO	
					.EXTRN	EDT\$-NOFILSPC, EDT\$-RECTOOBIG	
					.EXTRN	STR\$COPY_R	
					.PSECT	_EDT\$CODE, NOWRT, SHR, PIC, 2	
					.ENTRY	EDT\$\$INCL_CMD, Save R2,R3,R4,R5,R6,R7,R8,-	0732
						R9,R10,R11	
					MOVAB	EDT\$\$FMT_MSG, R11	
					MOVAB	EDT\$\$FIOPN_ERR, R10	
					MOVAB	EDT\$\$CALLFIO, R9	
					MOVL	#EDT\$K_INCLUDE_FILE, R8	
					SUBL2	#12, SP	
					PUSHL	#34471936	0828
					MOVL	#34471936, FILE_DESC	0830
					CLRL	RECORD_TOO_BIG	0823
					CLRL	RHB_DESC+4	0827
					CLRL	FILE_DESC+4	0829
					MOVL	EDT\$\$A_EXE_CURCMD, R0	0831
					PUSHL	12(R0)	
					MOVL	8(R0), STRADR	0832
					TSTL	STRLEN	0837
					BNEQ	1\$	
					PUSHL	#EDT\$ NOFILSPC	0840
					CALLS	#1, EDT\$\$FMT_MSG	
					RET		0841
					PUSHL	STRADR	0844
					PUSHAB	STRLEN	
					PUSHAB	FILE_DESC	
					CALLS	#3, STR\$COPY_R	
					PUSHAB	RHB_DESC	0845
					PUSHAB	FILE_DESC	
					PUSHL	R8	
					PUSHL	#EDT\$K_OPEN_INPUT	
					CALLS	#4, EDT\$\$CALLFIO	
					MOVL	R0, STATUS	
					BLBS	STATUS, 2\$	0847
					BRW	12\$	
					MOVL	#1, EDT\$\$G_RNG_FRSTLN	0853
					MOVL	EDT\$\$A_CUR_BUF, R0	0854
					MOVCL3	#14, (R0), EDT\$\$Z_RNG_ORIGPOS	
					MOVL	EDT\$\$A_EXE_CURCMD, R0	0856
					PUSHL	4(R0)	
					CALLS	#1, EDT\$\$RNG_REPOS	
					BLBS	R0, 3\$	
					RET		
					CALLS	#0, EDT\$\$START_INS	0861
					MOVL	#1, STATUS	0862
					BEQL	5\$	0864
					PUSHAB	RHB_DESC	0866
					PUSHAB	FILE_DESC	

		58	DD	000B9	PUSHL	R8			
		8F	DD	000BB	PUSHL	#EDTSK GET			
	69	04	FB	000C1	CALLS	#4, EDT\$\$CALLFIO			
	52	50	DO	000C4	MOVL	RO, GET			
0001827A	8F	52	D1	000C7	CMPL	GET, #98938			0871
		78	13	000CE	BEQL	11\$			
	0F	52	E8	000D0	BLBS	GET, 6\$			0873
		00	DD	000D3	PUSHL	EDT\$\$A_IO_FNAM			
		8F	DD	000D9	PUSHL	#EDTS_ERRINPFIL			
00000000G	6A	02	FB	000DF	CALLS	#2, EDT\$\$FIOPN_ERR			
	00	00	FB	000E2	CALLS	#0, EDT\$\$CHK_CC			0881
	27	50	E9	000E9	BLBC	RO, 8\$			
		7E	7C	000EC	CLRQ	-(SP)			0884
		58	DD	000EE	PUSHL	R8			
		8F	DD	000F0	PUSHL	#EDTSK CLOSE			
	69	04	FB	000F6	CALLS	#4, EDT\$\$CALLFIO			
	56	50	DO	000F9	MOVL	RO, STATUS			
	0C	56	E8	000FC	BLBS	STATUS, 7\$			0886
		0C	AE	9F 000FF	PUSHAB	FILE_DESC			
		8F	DD	00102	PUSHL	#EDTS_INCFILCLO			
00000000G	6A	02	FB	00108	CALLS	#2, EDT\$\$FIOPN_ERR			
	00	01	DO	0010B	MOVL	#1, EDT\$\$G_CC_DONE			0888
		04	04	00112	RET				0889
	00FF	8F	0C	AE B1 00113	CMPW	FILE_DESC, #255			0892
		03	1B	00119	BLEQU	9\$			
	57	01	DO	0011B	MOVL	#1, RECORD_TOO_BIG			
00000000G	00	00	9E	0011E	MOVAB	EDT\$\$T_LN_BUF, EDT\$\$A_SEL_POS			0894
	7E	0C	AE	3C 00129	MOVZWL	FILE_DESC -(SP)			0895
	00FF	8F	6E	B1 0012D	CMPW	(SP), #255			
		04	1B	00132	BLEQU	10\$			
	6E	FF	8F	9A 00134	MOVZBL	#255, (SP)			
		14	AE	DD 00138	PUSHL	FILE_DESC+4			
00000000G	00	02	FB	0013B	CALLS	#2, EDT\$\$INS_LN			
	56	50	DO	00142	MOVL	RO, STATUS			
		FF69	31	00145	BRW	4\$			0864
00000000G	00	00	FB	00148	CALLS	#0, EDT\$\$END_INS			0898
00000000G	00	00	FB	0014F	CALLS	#0, EDT\$\$STOP_WKINGMSG			0902
		7E	7C	00156	CLRQ	-(SP)			0903
		58	DD	00158	PUSHL	R8			
		8F	DD	0015A	PUSHL	#EDTSK CLOSE			
	69	04	FB	00160	CALLS	#4, EDT\$\$CALLFIO			
	56	50	DO	00163	MOVL	RO, STATUS			
	26	56	E8	00166	BLBS	STATUS, 15\$			0905
		0C	AE	9F 00169	PUSHAB	FILE_DESC			
		8F	DD	0016C	PUSHL	#EDTS_INCFILCLO			
		18	11	00172	BRB	14\$			
		0C	AE	9F 00174	PUSHAB	FILE_DESC			0913
08	00000000G	00	E9	00177	BLBC	EDT\$\$G_IOFI_NFND, 13\$			0915
	00000000G	8F	DD	0017E	PUSHL	#EDTS_INCFILNEX			
		06	11	00184	BRB	14\$			
		8F	DD	00186	PUSHL	#EDTS_INCFILOPN			
	6A	02	FB	0018C	CALLS	#2, EDT\$\$FIOPN_ERR			
	10	57	E9	0018F	BLBC	RECORD_TOO_BIG, 16\$			0925
		8F	DD	00192	PUSHL	#EDTS_RECTOOBIG			0928
	6B	01	FB	00198	CALLS	#1, EDT\$\$FMT_MSG			
00000000G	00	01	DO	0019B	MOVL	#1, EDT\$\$G_TXT_ONSCR			0929
		0C	AE	9F 001A2	PUSHAB	FILE_DESC			0938

: R

:  
:

EDT\$LINCL  
V04-000

EDT\$LINCL - INCLUDE line-mode command  
EDT\$\$INCL\_CMD - Line mode INCLUDE command

L 2  
16-Sep-1984 00:51:24  
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]LINCL.BLI;1

Page 10  
(3)

00000000G 00 01 FB 001A5  
04 AE 9F 001AC  
00000000G 00 01 FB 001AF  
04 0C1B6

CALLS #1, STR\$FREE1\_DX  
PUSHAB RMB\_DESC  
CALLS #1, STR\$FREE1\_DX  
RET

:  
: 0939  
:  
: 0942

: Routine Size: 439 bytes, Routine Base: \_EDT\$CODE + 0000

: 344 0943 1  
: 345 0944 1 !<BLF/PAGE>

EDT  
V04

SR  
L  
C

EDT\$LINCL  
V04-000

EDT\$LINCL - INCLUDE line-mode command  
EDT\$\$INCL\_CMD - Line mode INCLUDE command

M 2  
16-Sep-1984 00:51:24  
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[EDT.SRC]LINCL.BLI;1

Page 11  
(4)

: 347 0945 1 END  
: 348 0946 1  
: 349 0947 0 ELUDOM

! of module EDT\$LINCL

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	439	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	84	22	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	7	0	581	00:04.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:LINCL/OBJ=OBJ\$:LINCL MSRC\$:LINCL.BLI/UPDATE=(ENH\$:LINCL)

: Size: 439 code + 0 data bytes  
: Run Time: 00:26.7  
: Elapsed Time: 00:38.4  
: Lines/CPU Min: 2132  
: Lexemes/CPU-Min: 7281  
: Memory Used: 162 pages  
: Compilation Complete

