

[illegible]

EXE

Mod

ED 1

ED

ED

ED

ED

ED1

ED

ED

ED
EDED
EDSYS
LIB

L18

.....

```
RRRRRRRR      EEEEEEEEE  AAAAAA      JJ      000000  UU      UU      TTTTTTTTTT  EEEEEEEEE  XX      XX
RRRRRRRR      EEEEEEEEE  AAAAAA      JJ      000000  UU      UU      TTTTTTTTTT  EEEEEEEEE  XX      XX
RR      RR      EE      AA      AA      JJ      00      00      UU      UU      TT      EE      XX      XX
RR      RR      EE      AA      AA      JJ      00      00      UU      UU      TT      EE      XX      XX
RR      RR      EE      AA      AA      JJ      00      00      UU      UU      TT      EE      XX      XX
RRRRRRRR      EEEEEEEEE  AA      AA      JJ      00      00      UU      UU      TT      EE      XX      XX
RRRRRRRR      EEEEEEEEE  AA      AA      JJ      00      00      UU      UU      TT      EE      XX      XX
RR      RR      EE      AAAAAAAAAA  JJ      JJ      00      00      UU      UU      TT      EE      XX      XX
RR      RR      EE      AAAAAAAAAA  JJ      JJ      00      00      UU      UU      TT      EE      XX      XX
RR      RR      EE      AA      AA      JJ      JJ      00      00      UU      UU      TT      EE      XX      XX
RR      RR      EE      AA      AA      JJ      JJ      00      00      UU      UU      TT      EE      XX      XX
RR      RR      EEEEEEEEE  AA      AA      JJJJJJ  JJJJJJ  000000  UUUUUUUUUU  TT      EEEEEEEEE  XX      XX
RR      RR      EEEEEEEEE  AA      AA      JJJJJJ  JJJJJJ  000000  UUUUUUUUUU  TT      EEEEEEEEE  XX      XX
                                         ....
                                         ....
                                         ....
                                         ....
```

```
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
```



```

0001 0 XTITLE 'EDT$REAJOUTEX - read a text record from the journal file'
0002 0 MODULE EDT$REAJOUTEX ( ! Read a text record from the journal file
0003 0 IDENT = 'V04-000' ! File: REAJOUTEX.BLI Edit: JBS1015
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 Read a text record from the journal file. All other records are
0037 1 processed in this module.
0038 1
0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0040 1
0041 1 AUTHOR: John Sauter, CREATION DATE: June 22, 1981
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 1-001 - Original. JBS 22-Jun-1981
0046 1 1-002 - Change the format of the journal file to make it compatible
0047 1 with EDT v2.0. JBS 07-Jul-1981
0048 1 1-003 - Fix the module name and title. JBS 07-Aug-1981
0049 1 1-004 - Read from the journal file, not the command file. JBS 16-Aug-1981
0050 1 1-005 - Handle control C records. JBS 21-Dec-1981
0051 1 1-006 - Debug control C record handling. JBS 24-Dec-1981
0052 1 1-007 - Revise control C data names. JBS 29-Dec-1981
0053 1 1-008 - Use two words for control C counters.
0054 1 1-009 - Convert to use EDT$FILEIO. STS 18-Jan-1982
0055 1 1-010 - Pass RHB descriptor to fileio. STS 25-Jan-1982
0056 1 1-011 - Deallocate dynamic descriptors. STS 10-Feb-1982
0057 1 1-012 - Add literals for callable EDT. STS 10-Mar-1982

```


EDTSREAJOUTEX
V04-000

EDTSREAJOUTEX - read a text record from the jou

E 10
16-Sep-1984 01:27:14
14-Sep-1984 12:24:20

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]REAJOUTEX.BLI;1

Page 2
(1)

:	58	0058	1	:	1-013	- Remove reference to EDTSSRD IFI.	JBS 15-Mar-1982
:	59	0059	1	:	1-014	- Allow 0-length records.	JBS 01-Apr-1982
:	60	0060	1	:	1-015	- Use symbols instead of magic numbers when journaling control C.	JBS 24-May-1982
:	61	0061	1	:	--		
:	62	0062	1	:			

```

: 64      0063 1 %SBTTL 'Declarations'
: 65      0064 1
: 66      0065 1  TABLE OF CONTENTS:
: 67      0066 1
: 68      0067 1
: 69      0068 1  REQUIRE 'EDTSRC:TRAROUNAM';
: 70      0507 1
: 71      0508 1  FORWARD ROUTINE
: 72      0509 1    EDT$SRD_JOUTXT;
: 73      0510 1
: 74      0511 1
: 75      0512 1  INCLUDE FILES:
: 76      0513 1
: 77      0514 1
: 78      0515 1  REQUIRE 'EDTSRC:EDTREQ';
: 79      0650 1
: 80      L 0651 1  %IF %BLISS (BLISS32)
: 81      0652 1  %THEN
: 82      0653 1
: 83      0654 1  REQUIRE 'EDTSRC:SYSSYM';
: 84      0684 1
: 85      0685 1  %FI
: 86      0686 1
: 87      0687 1
: 88      0688 1  MACROS:
: 89      0689 1
: 90      0690 1    NONE
: 91      0691 1
: 92      0692 1  EQUATED SYMBOLS:
: 93      0693 1
: 94      0694 1
: 95      0695 1  EXTERNAL LITERAL
: 96      0696 1    EDT$K_GET,
: 97      0697 1    EDT$K_JOURNAL_FILE;
: 98      0698 1
: 99      0699 1
: 100     0700 1  OWN STORAGE:
: 101     0701 1
: 102     0702 1    NONE
: 103     0703 1
: 104     0704 1  EXTERNAL REFERENCES:
: 105     0705 1
: 106     0706 1  In the routine
```



```
108 0707 1 %SBTTL 'EDTSSRD_JOUTXT - read a text record from the journal file'
109 0708 1
110 0709 1 GLOBAL ROUTINE EDTSSRD_JOUTXT (      ! Read a text record from the journal file
111 0710 1     INPUT_BUFFER,                    ! Address of input buffer
112 0711 1     INPUT_BUFFER_LEN                ! Length of record read
113 0712 1 ) =
114 0713 1
115 0714 1 ++
116 0715 1 FUNCTIONAL DESCRIPTION:
117 0716 1
118 0717 1     Read records from the journal file until we find a text record. Return that
119 0718 1     record to the caller. The other kind of record, the control C record, is
120 0719 1     processed in this routine by storing its value in EDTSSG_CC_CNT2.
121 0720 1
122 0721 1 FORMAL PARAMETERS:
123 0722 1
124 0723 1     INPUT_BUFFER                        The address of the input buffer. The text record is
125 0724 1                                         stored in this buffer.
126 0725 1     INPUT_BUFFER_LEN                    The length of the text record read is stored here.
127 0726 1
128 0727 1 IMPLICIT INPUTS:
129 0728 1
130 0729 1     EDTSSG_RCOV_MOD
131 0730 1     EDTSSG_CC_CNT1_LO
132 0731 1     EDTSSG_CC_CNT1_HI
133 0732 1     EDTSSG_CC_CNT2_LO
134 0733 1     EDTSSG_CC_CNT2_HI
135 0734 1     EDTSSG_CC_FLAG
136 0735 1
137 0736 1 IMPLICIT OUTPUTS:
138 0737 1
139 0738 1     EDTSSG_CC_CNT1_LO
140 0739 1     EDTSSG_CC_CNT1_HI
141 0740 1     EDTSSG_CC_CNT2_LO
142 0741 1     EDTSSG_CC_CNT2_HI
143 0742 1     EDTSSG_CC_FLAG
144 0743 1
145 0744 1 ROUTINE VALUE:
146 0745 1
147 0746 1     1 = text record read, 0 = error from reading file
148 0747 1
149 0748 1 SIDE EFFECTS:
150 0749 1
151 0750 1     NONE
152 0751 1
153 0752 1 --
154 0753 1
155 0754 2 BEGIN
156 0755 2
157 0756 2 EXTERNAL ROUTINE
158 0757 2     EDTSSCALLF10;
159 0758 2
160 L 0759 2 %IF %BLISS (BLISS32)
161 0760 2 %THEN
162 0761 2
163 0762 2 EXTERNAL ROUTINE
164 0763 2     STR$FREE1_DX;
```

```

165 0764 2
166 0765 2 %FI
167 0766 2
168 0767 2 EXTERNAL
169 0768 2 EDT$G_CC_CNT1_LO,
170 0769 2 EDT$G_CC_CNT1_HI,
171 0770 2 EDT$G_CC_CNT2_LO,
172 0771 2 EDT$G_CC_CNT2_HI,
173 0772 2 EDT$G_CC_FLAG,
174 0773 2 EDT$G_RCOV_MOD;
175 0774 2
176 0775 2 LOCAL
177 0776 2 FILE_DESC : BLOCK [8, BYTE],
178 0777 2 RHB_DESC : BLOCK [8, BYTE],
179 0778 2 INPUT_FILE_STATUS,
180 0779 2 LOCAL_INPUT_BUFFER : REF VECTOR [, BYTE],
181 0780 2 LOCAL_INPUT_BUFFER_LEN;
182 0781 2
183 L 0782 2 %IF %BLISS (BLISS32)
184 0783 2 %THEN
185 0784 2 RHB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
186 0785 2 RHB_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
187 0786 2 FILE_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
188 0787 2 FILE_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
189 0788 2 %FI
190 0789 2
191 0790 2 RHB_DESC [DSC$A_POINTER] = 0;
192 0791 2 RHB_DESC [DSC$W_LENGTH] = 0;
193 0792 2 FILE_DESC [DSC$A_POINTER] = 0;
194 0793 2 FILE_DESC [DSC$W_LENGTH] = 0;
195 0794 2
196 0795 2 !+ This routine is only called if we are recovering.
197 0796 2 !-
198 0797 2 ASSERT (.EDT$G_RCOV_MOD);
199 0798 2 !+
200 0799 2 Read from the journal file until we get a text record.
201 0800 2 !-
202 0801 2
203 0802 2 DO
204 0803 2 BEGIN
205 0804 2 INPUT_FILE_STATUS = EDT$CALLFIO (EDT$K_GET, EDT$K_JOURNAL_FILE, FILE_DESC, RHB_DESC);
206 0805 2 LOCAL_INPUT_BUFFER = .FILE_DESC [DSC$A_POINTER];
207 0806 2 LOCAL_INPUT_BUFFER_LEN = .FILE_DESC [DSC$W_LENGTH];
208 0807 2
209 0808 2 IF .INPUT_FILE_STATUS
210 0809 2 THEN
211 0810 2 BEGIN
212 0811 2
213 0812 2 IF (.LOCAL_INPUT_BUFFER_LEN EQL 0)
214 0813 2 THEN
215 0814 2 BEGIN
216 0815 2 !+
217 0816 2 We have a 0-length record. Treat it as a text record.
218 0817 2 !-
219 0818 2 .INPUT_BUFFER_LEN = 0;
220 0819 2 !+
221 0820 2 On VMS deallocate all dynamic descriptors used
```



```
222 0821 5 !-
223 0822 5
224 L 0823 5 %IF %BLISS (BLISS32)
225 0824 5 %THEN
226 0825 5 STR$FREE1_DX (FILE_DESC);
227 0826 5 STR$FREE1_DX (RHB_DESC);
228 0827 5 %FI
229 0828 5
230 0829 5 RETURN (.INPUT_FILE_STATUS);
231 0830 5 END
232 0831 4 ELSE
233 0832 5 BEGIN
234 0833 5
235 0834 6 IF (.LOCAL_INPUT_BUFFER [0] NEQ JOUT_REC_ESC)
236 0835 5 THEN
237 0836 6 BEGIN
238 0837 6 !+
239 0838 6 We have a text record, move the record to the caller's buffer and return.
240 0839 6 !-
241 0840 6 ASSERT (.LOCAL_INPUT_BUFFER_LEN LEQ 256);
242 0841 6 CH$MOVE (.LOCAL_INPUT_BUFFER_LEN, LOCAL_INPUT_BUFFER [0], .INPUT_BUFFER);
243 0842 6 .INPUT_BUFFER_LEN = .LOCAL_INPUT_BUFFER_LEN;
244 0843 6 !+
245 0844 6 On VMS deallocate all dynamic descriptors used
246 0845 6 !-
247 0846 6
248 L 0847 6 %IF %BLISS (BLISS32)
249 0848 6 %THEN
250 0849 6 STR$FREE1_DX (FILE_DESC);
251 0850 6 STR$FREE1_DX (RHB_DESC);
252 0851 6 %FI
253 0852 6
254 0853 6 RETURN (.INPUT_FILE_STATUS);
255 0854 6 END
256 0855 5 ELSE
257 0856 6 BEGIN
258 0857 6 !+
259 0858 6 This must be a control C record. Arrange for a simulated control C trap
260 0859 6 after the specified number of tests of the control C flag.
261 0860 6 !-
262 0861 6 ASSERT (.LOCAL_INPUT_BUFFER [0] EQL JOUT_REC_ESC);
263 0862 6 ASSERT (.LOCAL_INPUT_BUFFER [1] EQL CC_REC_FLAG);
264 0863 6 ASSERT ((.LOCAL_INPUT_BUFFER_LEN) EQL CC_REC_SIZE);
265 0864 6 ASSERT (.EDT$$G_CC_CNT1_LO EQL 0);
266 0865 6 ASSERT (.EDT$$G_CC_CNT1_HI EQL 0);
267 0866 6 ASSERT (.EDT$$G_CC_CNT2_LO EQL 0);
268 0867 6 ASSERT (.EDT$$G_CC_CNT2_HI EQL 0);
269 0868 6 ASSERT (.EDT$$G_CC_FLAG EQL 0);
270 0869 6 EDT$$CPY_MEM (2, LOCAL_INPUT_BUFFER [2], EDT$$G_CC_CNT2_LO);
271 0870 6 EDT$$CPY_MEM (2, LOCAL_INPUT_BUFFER [4], EDT$$G_CC_CNT2_HI);
272 0871 6 ASSERT (.EDT$$G_CC_CNT2_LO GEQ 0);
273 0872 6 ASSERT (.EDT$$G_CC_CNT2_LO LEQ CC_CTR_MAX);
274 0873 6 ASSERT (.EDT$$G_CC_CNT2_HI GEQ 0);
275 0874 6 ASSERT (.EDT$$G_CC_CNT2_HI LEQ CC_CTR_MAX);
276 0875 6 EDT$$G_CC_FLAG = 1;
277 0876 5 END;
278 0877 5
```



```
279      0878 4      END;
280      0879 4
281      0880      END
282      0881      ELSE
283      0882      +
284      0883      - No more records in the journal file, return failure.
285      0884
286      0885      BEGIN
287      0886      RETURN (.INPUT_FILE_STATUS);
288      0887      END;
289      0888
290      0889      END
291      0890      UNTIL 0;
292      0891
293      0892      +
294      0893      - Control cannot reach this point.
295      0894
296      0895      ASSERT (0);
297      0896      RETURN (0);
298      0897      1      END;
```

! of routine EDT\$\$RD_JOUTXT

```
.TITLE EDT$REAJOUTEX EDT$REAJOUTEX - read a text recor
d from the jou
.IDENT \V04-000\
```

```
.EXTRN EDT$K_GET, EDT$K_JOURNAL_FILE
.EXTRN EDT$$CALLFIO, STR$FREE1_DX
.EXTRN EDT$$G_CC_CNT1_LO
.EXTRN EDT$$G_CC_CNT1_HI
.EXTRN EDT$$G_CC_CNT2_LO
.EXTRN EDT$$G_CC_CNT2_HI
.EXTRN EDT$$G_CC_FLAG, EDT$$G_RCOV_MOD
.EXTRN EDT$$INTER_ERR
```

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

```
.ENTRY EDT$$RD_JOUTXT, Save R2,R3,R4,R5,R6,R7,R8,- : 0709
R9,R10,R11
MOVAB EDT$$G_CC_CNT2_HI, R11
MOVAB EDT$$G_CC_CNT2_LO, R10
MOVAB EDT$$INTER_ERR, R9
SUBL2 #12, SP
PUSHL #34471936 : 0791
MOVL #34471936, FILE_DESC : 0793
CLRL RHB_DESC+4 : 0790
CLRL FILE_DESC+4 : 0792
BLBS EDT$$G_RCOV_MOD, 1$ : 0797
CALLS #0, EDT$$INTER_ERR
PUSHL SP : 0804
PUSHAB FILE_DESC
PUSHL #EDT$K_JOURNAL_FILE
PUSHL #EDT$K_GET
CALLS #4, EDT$$CALLFIO
MOVL R0, INPUT_FILE_STATUS
MOVL FILE_DESC+4, LOCAL_INPUT_BUFFER : 0805
MOVZWL FILE_DESC, LOCAL_INPUT_BUFFER_LEN : 0806
```

```
OFFC 00000
5B 00000000G 00 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 00 9E 00010
5E 0C C2 00017
08 AE 020E0000 8F DD 0001A
04 AE D4 00020
0C AE D4 00028
03 00000000G 00 E8 0002E
69 00 FB 00035
0C AE 9F 0003A 1$:
00000000G 8F DD 0003D
00000000G 8F DD 00043
00000000G 00 04 FB 00049
58 50 D0 00050
56 0C AE D0 00053
57 08 AE 3C 00057
```

35	58	E9	0005B	BLBC	INPUT_FILE_STATUS, 5\$	: 0808		
	05	12	0005E	BNEQ	2\$	: 0812		
	08	BC	D4	00060	CLRL	@INPUT_BUFFER_LEN	: 0818	
	1B	11	00063	BRB	4\$	: 0825		
FF	8F	66	91	00065	2\$: CMPB	(LOCAL_INPUT_BUFFER), #255	: 0834	
	2A	13	00069	BEQL	6\$			
00000100	8F	57	D1	0006B	CMPL	LOCAL_INPUT_BUFFER_LEN, #256	: 0840	
	03	15	00072	BLEQ	3\$			
	69	00	FB	00074	CALLS	#0, EDT\$\$INTER_ERR		
04	BC	66	57	28	00077	3\$: MOVCL	LOCAL_INPUT_BUFFER_LEN, -	: 0841
						(LOCAL_INPUT_BUFFER), @INPUT_BUFFER_LEN		
	08	BC	57	D0	0007C	MOVL	LOCAL_INPUT_BUFFER_LEN, @INPUT_BUFFER_LEN	: 0842
		08	AE	9F	00080	4\$: PUSHAB	FILE_DESC	: 0849
00000000G	00	01	FB	00083	CALLS	#1, STR\$FREE1_DX		
		5E	DD	0008A	PUSHL	SP	: 0850	
00000000G	00	01	FB	0008C	CALLS	#1, STR\$FREE1_DX		
		78	11	00093	5\$: BRB	18\$	: 0853	
	01	01	A6	91	00095	6\$: CMPB	1(LOCAL_INPUT_BUFFER), #1	: 0862
			03	13	00099	BEQL	7\$	
69		00	FB	0009B	CALLS	#0, EDT\$\$INTER_ERR		
06		57	D1	0009E	7\$: CMPL	LOCAL_INPUT_BUFFER_LEN, #6	: 0863	
		03	13	000A1	BEQL	8\$		
69		00	FB	000A3	CALLS	#0, EDT\$\$INTER_ERR		
	00000000G	00	D5	000A6	8\$: TSTL	EDT\$\$G_CC_CNT1_LO	: 0864	
		03	13	000AC	BEQL	9\$		
69		00	FB	000AE	CALLS	#0, EDT\$\$INTER_ERR		
	00000000G	00	D5	000B1	9\$: TSTL	EDT\$\$G_CC_CNT1_HI	: 0865	
		03	13	000B7	BEQL	10\$		
69		00	FB	000B9	CALLS	#0, EDT\$\$INTER_ERR		
		6A	D5	000BC	10\$: TSTL	EDT\$\$G_CC_CNT2_LO	: 0866	
		03	13	000BE	BEQL	11\$		
69		00	FB	000C0	CALLS	#0, EDT\$\$INTER_ERR		
		6B	D5	000C3	11\$: TSTL	EDT\$\$G_CC_CNT2_HI	: 0867	
		03	13	000C5	BEQL	12\$		
69		00	FB	000C7	CALLS	#0, EDT\$\$INTER_ERR		
	00000000G	00	D5	000CA	12\$: TSTL	EDT\$\$G_CC_FLAG	: 0868	
		03	13	000D0	BEQL	13\$		
69		00	FB	000D2	CALLS	#0, EDT\$\$INTER_ERR		
		A6	B0	000D5	13\$: MOVW	2(LOCAL_INPUT_BUFFER), EDT\$\$G_CC_CNT2_LO	: 0869	
		6B	04	B0	000D9	MOVW	4(LOCAL_INPUT_BUFFER), EDT\$\$G_CC_CNT2_HI	: 0870
		6A	D5	000DD	TSTL	EDT\$\$G_CC_CNT2_LO	: 0871	
		03	18	000DF	BGEQ	14\$		
00007530	69	00	FB	000E1	CALLS	#0, EDT\$\$INTER_ERR		
	8F	6A	D1	000E4	14\$: CMPL	EDT\$\$G_CC_CNT2_LO, #30000	: 0872	
		03	15	000EB	BLEQ	15\$		
	69	00	FB	000ED	CALLS	#0, EDT\$\$INTER_ERR		
		6B	D5	000F0	15\$: TSTL	EDT\$\$G_CC_CNT2_HI	: 0873	
		03	18	000F2	BGEQ	16\$		
00007530	69	00	FB	000F4	CALLS	#0, EDT\$\$INTER_ERR		
	8F	6B	D1	000F7	16\$: CMPL	EDT\$\$G_CC_CNT2_HI, #30000	: 0874	
		03	15	000FE	BLEQ	17\$		
	69	00	FB	00100	CALLS	#0, EDT\$\$INTER_ERR		
	00000000G	00	01	D0	00103	17\$: MOVL	#1, EDT\$\$G_CC_FLAG	: 0875
		FF2B	31	0010A	BRW	1\$	: 0808	
		58	D0	0010D	18\$: MOVL	INPUT_FILE_STATUS, R0	: 0886	
	50	04	00110	RET			: 0897	

EDT\$REAJOUTEX EDT\$REAJOUTEX - read a text record from the jou L 10
V04-000 EDT\$SRD_JOUTXT - read a text record from the j 16-Sep-1984 01:27:14
14-Sep-1984 12:24:20

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]REAJOUTEX.BLI;1

Page 9
(3)

; Routine Size: 273 bytes, Routine Base: _EDT\$CODE + 0000

: 299 0898 1
: 300 0899 1 !<BLF/PAGE>

EDT\$REAJOUTEX
V04-000

EDT\$REAJOUTEX - read a text record from the jou
EDT\$SRD_JOUTXT - read a text record from the j

M 10

16-Sep-1984 01:27:14
14-Sep-1984 12:24:20

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]REAJOUTEX.BLI;1

Page 10
(4)

: 302 0900 1 END
: 303 0901 1
: 304 0902 0 ELUDOM

! of module EDT\$REAJOUTEX

PSECT SUMMARY

Name Bytes Attributes
_EDT\$CODE 273 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	5	1	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	6	0	581	00:04.1

COMMAND QUALIFIERS

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

: Size: 273 code + 0 data bytes
: Run Time: 00:24.9
: Elapsed Time: 00:37.1
: Lines/CPU Min: 2171
: Lexemes/CPU-Min: 8874
: Memory Used: 154 pages
: Compilation Complete

0138

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY