

[illegible]

```

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

```



```
0001 0 %TITLE 'EDT$WFREAINP - read a line from the input file'
0002 0 MODULE EDT$WFREAINP ( ! Read a line from the input file
0003 0 IDENT = 'V04-000' ! File: WFREAINP.BLI Edit: JBS1036
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 line from the input file.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: October 16, 1978
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 23-Feb-1981. This module was created by
0045 1 extracting routine READ_INPUT from module EDTWF.
0046 1 1-002 - Regularize headers. JBS 19-Mar-1981
0047 1 1-003 - Change EOB_LINE to EDT$EZ_EOB_LN. JBS 31-Mar-1981
0048 1 1-004 - Remove division from line number calculations. SMB 14-Jan-1982
0049 1 1-005 - Convert i/o to use edt$fileio. STS 15-Jan-1982
0050 1 1-006 - Add capability for 15 digit line numbers. SMB 19-Jan-1982
0051 1 1-007 - pass header info by descriptor. STS 20-Jan-1982
0052 1 1-008 - fix bugs related to 15-digit line numbers. SMB 22-Jan-1982
0053 1 1-009 - check for zero length descriptor. STS 22-Jan-1982
0054 1 1-010 - Remove a MOVELINE for original line numbers. SMB 26-Jan-1982
0055 1 1-011 - Remove original line numbers. SMB 28-Jan-1982
0056 1 1-012 - Fix bug in line number assignment for sequenced files. SMB 30-Jan-1982
0057 1 1-013 - Add a range check for line number too large. SMB 1-Feb-1982
```

```
58 0058 1 1-014 - Pass address of io_rhb to file routines. STS 03-Feb-1982
59 0059 1 1-015 - Close input file if line numbers get too large. SMB 06-Feb-1982
60 0060 1 1-016 - Deallocate dynamic descriptors. STS 10-Feb-1982
61 0061 1 1-017 - Add literals for callable parameters. STS 08-Mar-1982
62 0062 1 1-018 - Remove reference to EDT$SRD_IF1. JBS 15-Mar-1982
63 0063 1 1-019 - Clear input file open flag when closing. STS 07-Apr-1982
64 0064 1 1-020 - Print a message on a CLOSE error. JBS 12-Apr-1982
65 0065 1 1-021 - Check for records longer than 255 bytes. JBS 02-Jun-1982
66 0066 1 1-022 - Close the input file on EOF. JBS 11-Jun-1982
67 0067 1 1-023 - Close the input file on reaching the end of the MAIN buffer only! JBS 14-Jun-1982
68 0068 1 1-024 - Print a message on Get errors and try again. STS 21-Jul-1982
69 0069 1 1-025 - Don't check RMS status on 11's since error message has already
70 0070 1 gone out. STS 22-Jul-1982
71 0071 1 1-026 - Print out the full RMS message rather than just 1 liner. STS 23-Jul-1982
72 0072 1 1-027 - Set the text on screen flag for RMS errors. SMB 28-Jul-1982
73 0073 1 1-028 - Change the flags set for error messages (again). SMB 17-Aug-1982
74 0074 1 1-029 - Modify to use new 48 bit macro. STS 01-Oct-1982
75 0075 1 1-030 - Don't clear EDT$SG_MSGFLG; setting EDT$SG_SCR_CHGD is sufficient. JBS 06-Oct-1982
76 0076 1 1-031 - Modify to use new compare macro. STS 20-Oct-1982
77 0077 1 1-032 - Use a different scrolling routine. SMB 21-Oct-1982
78 0078 1 1-033 - Set EDT$SG_SCR_REBUILD if we insert a line in the work file. JBS 01-Nov-1982
79 0079 1 1-034 - Modify edit 1-033 so we don't disturb the screen data base but don't
80 0080 1 have to rebuild the screen. JBS 01-Dec-1982
81 0081 1 1-035 - Note new value for EDT$SG_SCR_CHGD. JBS 02-Mar-1983
82 0082 1 1-036 - Improve message about RMS error. JBS 24-Mar-1983
83 0083 1 --
84 0084 1
```


EDT\$WFREAINP
V04-000

EDT\$WFREAINP - read a line from the input file
Declarations

N 12
16-Sep-1984 02:11:53
14-Sep-1984 12:25:41

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]WFREAINP.BLI;1
Page 3
(2)

```
: 86      0085 1 %SBTTL 'Declarations'
: 87      0086 1
: 88      0087 1 : TABLE OF CONTENTS:
: 89      0088 1 :
: 90      0089 1
: 91      0090 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 92      0529 1
: 93      0530 1 FORWARD ROUTINE
: 94      0531 1     EDT$SRD_ILN;
: 95      0532 1
: 96      0533 1 :
: 97      0534 1 : INCLUDE FILES:
: 98      0535 1 :
: 99      0536 1
: 100     0537 1 REQUIRE 'EDTSRC:EDTREQ';
: 101     0672 1
: 102     L 0673 1 %IF %BLISS (BLISS32)
: 103     0674 1 %THEN
: 104     0675 1
: 105     0676 1 REQUIRE 'EDTSRC:SYSSYM';
: 106     0706 1
: 107     0707 1 %FI
: 108     0708 1
: 109     0709 1 :
: 110     0710 1 : MACROS:
: 111     0711 1 :
: 112     0712 1 :     NONE
: 113     0713 1 :
: 114     0714 1 : EQUATED SYMBOLS:
: 115     0715 1 :
: 116     0716 1
: 117     0717 1 EXTERNAL LITERAL
: 118     0718 1     EDT$K_INPUT_FILE,
: 119     0719 1     EDT$K_GET,
: 120     0720 1     EDT$K_CLOSE;
: 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
```

```
131 0730 1 %SBTTL 'EDT$$RD_ILN - read a line from the input file'
132 0731 1
133 0732 1 GLOBAL ROUTINE EDT$$RD_ILN ! Read a line from the input file
134 0733 1 =
135 0734 1
136 0735 1 !++
137 0736 1 FUNCTIONAL DESCRIPTION:
138 0737 1
139 0738 1 This routine attempts to read a new line from the input file. If the
140 0739 1 INPUT_RAB field is zero, or if the read fails return 0 to indicate that
141 0740 1 there is no more input. If the read succeeds, give the line the next
142 0741 1 sequential number, and insert it at the end of the text buffer.
143 0742 1
144 0743 1 FORMAL PARAMETERS:
145 0744 1
146 0745 1 NONE
147 0746 1
148 0747 1 IMPLICIT INPUTS:
149 0748 1
150 0749 1 EDT$$G_SCR_LNS
151 0750 1 EDT$$A_CUR_BUF
152 0751 1 EDT$$L_WK_INSCNT
153 0752 1 EDT$$A_WK_LN
154 0753 1 EDT$$Z_EOB_LN
155 0754 1 EDT$$L_LNOO - L_LNO14
156 0755 1 EDT$$L_LNO_ZERO
157 0756 1 EDT$$L_LNO_BIG
158 0757 1
159 0758 1 IMPLICIT OUTPUTS:
160 0759 1
161 0760 1 EDT$$A_CUR_BUF
162 0761 1 EDT$$L_WK_INSCNT
163 0762 1 EDT$$L_IO_VFCHD
164 0763 1 EDT$$G_SCR_CHGD
165 0764 1
166 0765 1 ROUTINE VALUE:
167 0766 1
168 0767 1 1 = a line was read
169 0768 1 0 = no more input
170 0769 1
171 0770 1 SIDE EFFECTS:
172 0771 1
173 0772 1 NONE
174 0773 1
175 0774 1 --
176 0775 1
177 0776 2 BEGIN
178 0777 2
179 0778 2 EXTERNAL ROUTINE
180 0779 2 EDT$$FMT_MSG, ! Output a message
181 0780 2 EDT$$CALCFIO, ! Do file I/O
182 0781 2 EDT$$INS_LN : NOVALUE, ! Insert a line in the work file
183 0782 2 EDT$$RD_PrvLN; ! Read the previous line in the work file
184 0783 2
185 L 0784 2 %IF %BLISS (BLISS32)
186 0785 2 %THEN
187 0786 2
```



```
188 0787 2 EXTERNAL ROUTINE
189 0788 2 EDTSSFIOPN_ERR,
190 0789 2 EDTSSOUT_FMTBUF,
191 0790 2 EDTSSSC_FULLSCLL,
192 0791 2 EDTSSSTOP_WKINGMSG,
193 0792 2 EDTSSMSG_TOSTR,
194 0793 2 EDTSTI_INPCH,
195 0794 2 EDTSSC_NONREVID,
196 0795 2 STR$FREE1_DX;
197 0796
198 0797 2 %FI
199 0798
200 0799 2 EXTERNAL
201 0800 2 EDTSSG_SCR_LNS,
202 0801 2 EDTSSA_CUR_BUF : REF TBCB_BLOCK,
203 0802 2 EDTSSL_WK_INSCNT : LN_BLOCK,
204 0803 2 EDTSSG_INPUT_OPN,
205 0804 2 EDTSSL_LNO_ZERO : LN_BLOCK,
206 0805 2 EDTSSL_LNO_BIG : LN_BLOCK,
207 0806 2 EDTSSA_WK_CN : REF [IN_BLOCK,
208 0807 2 EDTSSZ_EOB_LN,
209 0808 2 EDTSSL_IO_VFCHD : WORD,
210 0809 2 EDTSSL_LN_PREV : WORD,
211 0810 2 EDTSSL_LN_INCR : LN_BLOCK,
212 0811 2 EDTSSL_LNDO : LNOVECTOR [14],
213 0812 2 EDTSSG_SCR_REBUILD;
214 0813
215 L 0814 2 %IF %BLISS (BLISS32)
216 0815 2 %THEN
217 0816
218 0817 2 EXTERNAL
219 0818 2 EDTSSA_IO_FNAM,
220 0819 2 EDTSSG_EDIT_MOD,
221 0820 2 EDTSSG_SCR_CHGD,
222 0821 2 EDTSSG_MSGFLG;
223 0822
224 0823 2 %FI
225 0824
226 0825 2 MESSAGES ((MAXINPLIN, PRERETCON, ERRINPFIL, INPFILCLO, RECTOOBIG));
227 0826
228 0827 2 LOCAL
229 0828 2 RHB_DESC : BLOCK [8, BYTE],
230 0829 2 FILE_DESC : BLOCK [8, BYTE],
231 0830 2 MAX_CNO,
232 0831 2 GET,
233 0832 2 LINNO : LN_BLOCK,
234 0833 2 I,
235 0834 2 NEW_BUKT,
236 0835 2 SAVE_REBUILD;
237 0836
238 L 0837 2 %IF %BLISS (BLISS32)
239 0838 2 %THEN
240 0839 2 RHB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
241 0840 2 RHB_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
242 0841 2 FILE_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
243 0842 2 FILE_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
244 0843 2 %FI
```

```
! Print file name with error message
! Output the format buffer
! Set for full-screen scrolling
! Turn off the 'working' message
! Translate a message to a string
! Input a character
! Set to non-reverse video character attributes

! Number of screen lines
! Current text buffer control block
! The count of inserted lines
! input file open flag
! Zero in 48-bit repr.
! 2814749767.00000
! Pointer to current line

! Record header buffer for fixed line nos.
! Previous VFC line number
! Input line number increment

! 1 = rebuild screen data base from the work file

! File name
! Edit mode: line mode or change mode
! 1 = screen must be repainted
! 1 = there is a message on the screen
```



```
245 0844 2
246 0845 RHB_DESC [DSC$A_POINTER] = 0;
247 0846 RHB_DESC [DSC$W_LENGTH] = 0;
248 0847 FILE_DESC [DSC$A_POINTER] = 0;
249 0848 FILE_DESC [DSC$W_LENGTH] = 0;
250 0849 MAX_CNO = 0;
251 0850 I = 2;
252 0851 STRING_DESC (RHB_DESC, I, EDT$$_IO_VFCHD);
253 0852
254 0853 + Try to read a record from input.
255 0854 -
256 0855
257 0856 IF (.EDT$$_CUR_BUF [TBCB_INPUT_RAB] NEQ 0)
258 0857 THEN
259 0858 BEGIN
260 0859 GET = EDT$$_CALLFIO (EDT$$_GET, EDT$$_INPUT_FILE, FILE_DESC, RHB_DESC);
261 0860
262 0861 + If the get failed but it was not end of file then print out an error
263 0862 message.
264 0863 -
265 0864
266 L 0865 %IF %BLISS (BLISS32)
267 0866 %THEN
268 0867
269 0868 IF (( NOT .GET) AND (.GET NEQ RMSS_EOF))
270 0869 THEN
271 0870
272 0871 IF (.EDT$$_G_EDIT_MOD EQL CHANGE_MODE)
273 0872 THEN
274 0873 BEGIN
275 0874 EDT$$_SC_FULLSCLL ();
276 0875 EDT$$_STOP_WKINGMSG ();
277 0876 EDT$$_FIOPN_ERR (EDT$$_ERRINPFIL, .EDT$$_IO_FNAM);
278 0877
279 0878 + Output the "Press return to continue"
280 0879 message and wait for ANY key to be pressed before
281 0880 refreshing the screen and returning to change mode
282 0881 -
283 0882 EDT$$_MSG_TOSTR (EDT$$_PRERETCON);
284 0883 EDT$$_OUT_FMTBUF ();
285 0884 EDT$$_TI_INPCH (I);
286 0885 EDT$$_SC_NONREVID ();
287 0886 EDT$$_G_SCR_CHGD = 1;
288 0887 EDT$$_G_MSGFLG = 1;
289 0888 END
290 0889 ELSE
291 0890 EDT$$_FIOPN_ERR (EDT$$_ERRINPFIL, .EDT$$_IO_FNAM);
292 0891
293 0892 %FI
294 0893
295 0894 IF .GET
296 0895 THEN
297 0896 BEGIN
298 0897
299 0898 + Determine the next line number
300 0899 -
301 0900
```



```

302 0901 5 IF (.RHB_DESC [DSC$W_LENGTH] NEQ 0)
303 0902 4 THEN
304 0903 3 BEGIN
305 0904 3 MOVELINE (EDT$SL_LNO_ZERO, LINNO);
306 0905 3 LINNO [LN_LO] = .RHB_DESC [DSC$A_POINTER];
307 0906 3
308 0907 6 IF (.EDT$SL_LN_PREV GEQU .LINNO [LN_LO]) !
309 0908 3 THEN
310 0909 3 ADDLINE (EDT$SL_LNOO [10], EDT$SL_LN_INCR, EDT$SL_LN_INCR, MAX_LNO);
311 0910 3
312 0911 6 IF (.MAX_LNO EQL 0)
313 0912 3 THEN
314 0913 6 BEGIN
315 0914 6 EDT$SL_LN_PREV = .LINNO [LN_LO];
316 0915 6 MULTILINE (EDT$SL_LNOO [5], [LINNO, LINNO]);
317 0916 6 ADDLINE (LINNO, EDT$SL_LN_INCR, EDT$A_CUR_BUF [TBCB_INPUT_LINE], MAX_LNO);
318 0917 6 END
319 0918 6
320 0919 3 END
321 0920 4 ELSE
322 P 0921 4 ADDLINE (EDT$A_CUR_BUF [TBCB_INPUT_LINE], EDT$SL_LNOO [5], !
323 0922 4 EDT$A_CUR_BUF [TBCB_INPUT_LINE], MAX_LNO);
324 0923 4
325 0924 4 !+
326 0925 4 | If the next line number is greater than the maximum allowed, stop reading
327 0926 4 | records and jump to code which displays end of buffer
328 0927 4 | -
329 0928 4
330 0929 5 IF ((CMPLNO (EDT$SL_LNO_BIG, EDT$A_CUR_BUF [TBCB_INPUT_LINE]) GEQ 0) AND (.MAX_LNO EQL 0))
331 0930 4 THEN
332 0931 3 BEGIN
333 0932 3 !+
334 0933 3 | The read succeeded and the line number is good. Insert the record at the end of the buffer.
335 0934 3 | Do not disturb the screen data base.
336 0935 3 | -
337 0936 3 SAVE REBUILD = .EDT$G_SCR_REBUILD;
338 0937 3 EDT$G_SCR_REBUILD = 1;
339 0938 3
340 0939 3 IF (.FILE_DESC [DSC$W_LENGTH] GTR 255) THEN EDT$FMT_MSG (EDT$_RECTOOBIG);
341 0940 3
342 0941 3 EDT$INS_LN (.FILE_DESC [DSC$A_POINTER], MIN (.FILE_DESC [DSC$W_LENGTH], 255));
343 0942 3 EDT$G_SCR_REBUILD = .SAVE_REBUILD;
344 0943 3 !+
345 0944 3 | In case we are doing an insert on another buffer (as in the
346 0945 3 | case of a CUT command), do not screw up the insert count.
347 0946 3 | -
348 0947 3 SUBLINE (NUMBER ONE, EDT$SL_WK_INSCNT);
349 0948 3 EDT$SRD_PRVLN ();
350 0949 3 MOVELINE (EDT$A_CUR_BUF [TBCB_INPUT_LINE], EDT$A_WK_LN [LIN_NUM]);
351 0950 3 !+
352 0951 3 | on VMS deallocate all dynamic descriptors used
353 0952 3 | -
354 0953 3
355 L 0954 3 %IF %BLISS (BLISS32)
356 0955 3 %THEN
357 0956 3 STR$FREE1_DX (FILE_DESC);
358 0957 3 STR$FREE1_DX (RHB_DESC);
```



```
359 0958 5 %FI
360 0959
361 0960
362 0961
363 0962
364 0963
365 0964
366 0965
367 0966
368 0967
369 0968
370 0969
371 0970
372 0971
373 0972
374 0973
375 0974
376 0975
377 0976
378 0977
379 0978
380 0979
381 0980
382 0981
383 0982
384 0983
385 0984
386 0985
387 0986
388 0987
389 0988
390 0989
391 0990
392 0991
393 0992
394 0993
395 0994
396 0995
397 0996
398 0997
399 0998
400 0999
401 1000
402 1001
403 1002
404 1003
405 1004
406 1005
407 1006
408 1007
409 1008
410 1009
411 1010
412 1011

!+
!- And return success
      RETURN (1);
      END
    ELSE
      BEGIN
        EDT$$FMT_MSG (EDT$_MAXINPLIN);
        IF ( NOT (EDT$$CALLFIO (EDT$K_CLOSE, EDT$K_INPUT_FILE, FILE_DESC, RHB_DESC)))
        THEN
          EDT$$FMT_MSG (EDT$_INPFILCLO);
          EDT$$G_INPUT_OPN = 0;
          END
      END;

!+
!- The read failed, close the file and indicate end of buffer.
      END;
      IF (.EDT$$G_INPUT_OPN AND (.EDT$$A_CUR_BUF [TBCB_INPUT_RAB] NEQ 0))
      THEN
        BEGIN
          IF ( NOT (EDT$$CALLFIO (EDT$K_CLOSE, EDT$K_INPUT_FILE, FILE_DESC, RHB_DESC)))
          THEN
            EDT$$FMT_MSG (EDT$_INPFILCLO);
            EDT$$G_INPUT_OPN = 0;
            END;
          EDT$$A_WK_LN = EDT$$Z_EOB_LN;
          EDT$$A_CUR_BUF [TBCB_CHAR_POS] = 0;
          EDT$$A_CUR_BUF [TBCB_INPUT_RAB] = 0;
          !+
          !- on VMS deallocate all dynamic descriptors used
        END
      !+
      !- %IF %BLISS (BLISS32)
      !+
      !- %THEN
      !+
      !- STR$FREE1_DX (FILE_DESC);
      !+
      !- STR$FREE1_DX (RHB_DESC);
      !+
      !- %FI
      !+
      !- And return failure
      !+
      !- RETURN (0);
      !+
      !- END;
```

! of routine EDT\$\$RD_ILN

.TITLE EDT\$WFREAINP EDT\$WFREAINP - read a line from th

EDTSWFREAINP
V04-000

EDTSWFREAINP - read a line from the input file
EDTSSRD_ILN - read a line from the input file

G 13

16-Sep-1984 02:11:53
14-Sep-1984 12:25:41

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

Page 9
(3)

**F

:

e input file

.IDENT \V04-000\

.EXTRN EDT\$K_INPUT_FILE
.EXTRN EDT\$K_GET, EDT\$K_CLOSE
.EXTRN EDT\$FMT_MSG, EDT\$CALLFIO
.EXTRN EDT\$INS_LN, EDT\$RD_PRVLN
.EXTRN EDT\$FIOPN_ERR, EDT\$OUT_FMTBUF
.EXTRN EDT\$SC_FUCLSCLL
.EXTRN EDT\$STOP_WKINGMSG
.EXTRN EDT\$MSG_TOSTR, EDT\$TI_INPCH
.EXTRN EDT\$SC_NONREVID
.EXTRN STR\$FREE1_DX, EDT\$G_SCR_LNS
.EXTRN EDT\$A_CUR_BUF, EDT\$SL_WR_INSCNT
.EXTRN EDT\$G_INPOT_OPN
.EXTRN EDT\$SL_LNO_ZERO
.EXTRN EDT\$SL_LNO_BIG, EDT\$A_WK_LN
.EXTRN EDT\$Z_EOB_LN, EDT\$SL_IO_VFCHD
.EXTRN EDT\$SL_LN_PREV, EDT\$C_LN_INCR
.EXTRN EDT\$SL_LN00, EDT\$G_SCR_REBUILD
.EXTRN EDT\$A_IO_FNAM, EDT\$G_EDIT_MOD
.EXTRN EDT\$G_SCR_CHGD
.EXTRN EDT\$G_MSGFLG, EDT\$MAXINPLIN
.EXTRN EDT\$PRERETCON, EDT\$ERRINPFIL
.EXTRN EDT\$INPFILCLO, EDT\$RECTOOBIG
.EXTRN STR\$COPY_R

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

.ENTRY EDT\$RD_ILN, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0732
R10,R11
MOVAB FIRST_WORD, R11
MOVAB EDT\$FMT_MSG, R10
MOVAB M1, R9
MOVAB EDT\$A_CUR_BUF, R8
MOVAB S2_UP, R7
SUBL2 #40, SP
MOVL #34471936, RHB_DESC ; 0846
MOVL #34471936, FILE_DESC ; 0848
CLRL RHB_DESC+4 ; 0845
CLRL FILE_DESC+4 ; 0847
CLRL MAX_LNO ; 0849
PUSHL #2 ; 0850
PUSHAB EDT\$SL_IO_VFCHD ; 0851
PUSHAB I
PUSHAB RHB_DESC
CALLS #3, STR\$COPY_R
MOVL EDT\$A_CUR_BUF, R0 ; 0856
TSTB 42(R0)
BNEQ 1\$
BRW 25\$
PUSHAB RHB_DESC ; 0859
PUSHAB FILE_DESC
PUSHL #EDT\$K_INPUT_FILE
PUSHL #EDT\$K_GET
CALLS #4, EDT\$CALLFIO
MOVL R0, GET

OFFC 00000

5B 00000000G 00 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 00 9E 00010
58 00000000G 00 9E 00017
57 00000000G 00 9E 0001E
20 5E 020E0000 28 C2 00025
18 AE 020E0000 8F D0 00028
AE 020E0000 8F D0 00030
24 AE D4 00038
1C AE D4 0003B
56 D4 0003E
02 DD 00040
00000000G 00 9F 00042
04 AE 9F 00048
2C AE 9F 0004B
00000000G 00 03 FB 0004E
50 68 D0 00055
2A A0 95 00058
03 12 0005B
0247 31 0005D
24 AE 9F 00060 1\$:
20 AE 9F 00063
00000000G 8F DD 00066
00000000G 8F DD 0006C
00000000G 00 04 FB 00072
52 50 D0 00079

Line	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
------	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

52		67	1A	AE	A1	00182	ADDW3	S1 UP, S2 UP, SAVES2	
		51	18	A0	B0	00187	MOVW	24(R0), SAVED	
12	A0	FA	A7	14	AE	C1	0018B	ADDL3	LINNO, EDT\$\$LN_INCR, 18(R0)
		16	A0	FE	A7	D0	00192	MOVL	EDT\$\$LN_INCR+4, 22(R0)
		16	A0	18	AE	D8	00197	ADWC	LINNO, -22(R0)
				16	11	0019C	BRB	12\$	
		50		68	D0	0019E	11\$:	MOVL	EDT\$\$A_CUR_BUF, R0
52	06	A9	18	A0	A1	001A1	ADDW3	24(R0), S2 UP, SAVES2	0922
		51	18	A0	B0	001A7	MOVW	24(R0), SAVED	
	12	A0		69	C0	001AB	ADDL2	EDT\$\$LN00+30, 18(R0)	
	16	A0	04	A9	D8	001AF	ADWC	EDT\$\$LN00+34, 22(R0)	
		52	18	A0	B1	001B4	12\$:	CMPW	24(R0), SAVES2
				04	12	001B8	BNEQ	13\$	
				56	D4	001BA	CLRL	MAX_LNO	
				03	11	001BC	BRB	14\$	
		56		01	D0	001BE	13\$:	MOVL	#1, MAX_LNO
50	18	A0		51	B0	001C1	14\$:	MOVW	SAVED, 24(R0)
		68		12	C1	001C5	15\$:	ADDL3	#18, EDT\$\$A_CUR_BUF, R0
		51	00000000G	00	3C	001C9	MOVZWL	HIGH 1, R1	0929
		51	04	A0	B1	001D0	CMPW	4(R0), R1	
				0E	1A	001D4	BGTRU	16\$	
				17	12	001D6	BNEQ	18\$	
		51	00000000G	00	D0	001D8	MOVL	LOW 1, R1	
		60		51	D1	001DF	CMPL	R1, -(R0)	
				05	1E	001E2	BGEQU	17\$	
		50		01	CE	001E4	16\$:	MNEGL	#1, R0
				09	11	001E7	BRB	19\$	
				04	12	001E9	17\$:	BNEQ	18\$
				50	D4	001EB	CLRL	R0	
				03	11	001ED	BRB	19\$	
		50		01	D0	001EF	18\$:	MOVL	#1, R0
				7F	19	001F2	19\$:	BLSS	23\$
				56	D5	001F4	TSTL	MAX_LNO	
				7B	12	001F6	BNEQ	23\$	
		52	00000000G	00	D0	001F8	MOVL	EDT\$\$G_SCR_REBUILD, SAVE_REBUILD	0936
	00000000G	00		01	D0	001FF	MOVL	#1, EDT\$\$G_SCR_REBUILD	0937
	00FF	8F	1C	AE	B1	00206	CMPW	FILE_DESC, #255	0939
				09	1B	0020C	BLEQU	20\$	
			00000000G	8F	DD	0020E	PUSHL	#EDT\$ RECTO0BIG	
		6A		01	FB	00214	CALLS	#1, EDT\$\$FMT MSG	
		7E	1C	AE	3C	00217	20\$:	MOVZWL	FILE_DESC, -(SP)
	00FF	8F		6E	B1	0021B	CMPW	(SP), #255	0941
				04	1B	00220	BLEQU	21\$	
		6E	FF	8F	9A	00222	MOVZBL	#255, (SP)	
			24	AE	DD	00226	21\$:	PUSHL	FILE_DESC+4
	00000000G	00		02	FB	00229	CALLS	#2, EDT\$\$INS_LN	
	00000000G	00		52	D0	00230	MOVL	SAVE_REBUILD, EDT\$\$G_SCR_REBUILD	0942
		50		6B	D0	00237	MOVL	FIRST_WORD, SAVE	0947
				6B	D7	0023A	DECL	FIRST_WORD	
		50		6B	D1	0023C	CMPL	FIRST_WORD, SAVE	
				03	1B	0023F	BLEQU	22\$	
			04	AB	B7	00241	DECW	NEXT_WORD	
	00000000G	00		00	FB	00244	22\$:	CALLS	#0, EDT\$\$RD PRVLN
		51		68	D0	0024B	MOVL	EDT\$\$A_CUR_BUF, R1	0948
		50	00000000G	00	D0	0024E	MOVL	EDT\$\$A_WK_CN, R0	0949
01	A0	12	A1	06	28	00255	MOVC3	#6, 18(R1), 1(R0)	
			1C	AE	9F	0025B	PUSHAB	FILE_DESC	0956

00000000G	00	01	FB	0025E	CALLS	#1, STR\$FREE1_DX	:	
		24	AE	9F 00265	PUSHAB	RHB_DESC	:	0957
00000000G	00	01	FB	00268	CALLS	#1, STR\$FREE1_DX	:	
	50	01	D0	0026F	MOVL	#1, R0	:	0963
		04	00272	RET			:	
	00000000G	8F	DD	00273	PUSHL	#EDT\$ MAXINPLIN	:	0967
	6A	01	FB	00279	CALLS	#1, EDT\$FMT_MSG	:	
		24	AE	9F 0027C	PUSHAB	RHB_DESC	:	0969
		20	AE	9F 0027F	PUSHAB	FILE_DESC	:	
	00000000G	8F	DD	00282	PUSHL	#EDT\$K_INPUT_FILE	:	
	00000000G	8F	DD	00288	PUSHL	#EDT\$K_CLOSE	:	
00000000G	00	04	FB	0028E	CALLS	#4, EDT\$CALLFIO	:	
	09	50	E8	00295	BLBS	R0, 24\$	:	
	00000000G	8F	DD	00298	PUSHL	#EDT\$ INPFILCLO	:	0971
	6A	01	FB	0029E	CALLS	#1, EDT\$FMT_MSG	:	
	00000000G	00	D4	002A1	CLRL	EDT\$G_INPUT_OPN	:	0973
	33	00	E9	002A7	BLBC	EDT\$G_INPUT_OPN, 27\$	:	0983
	50	68	D0	002AE	MOVL	EDT\$A_CUR_BUF, R0	:	
		2A	A0	95 002B1	TSTB	42(R0)	:	
		2B	13	002B4	BEQL	27\$	:	
		24	AE	9F 002B6	PUSHAB	RHB_DESC	:	0987
		20	AE	9F 002B9	PUSHAB	FILE_DESC	:	
	00000000G	8F	DD	002BC	PUSHL	#EDT\$K_INPUT_FILE	:	
	00000000G	8F	DD	002C2	PUSHL	#EDT\$K_CLOSE	:	
00000000G	00	04	FB	002C8	CALLS	#4, EDT\$CALLFIO	:	
	09	50	E8	002CF	BLBS	R0, 26\$	:	
	00000000G	8F	DD	002D2	PUSHL	#EDT\$ INPFILCLO	:	0989
	6A	01	FB	002D8	CALLS	#1, EDT\$FMT_MSG	:	
	00000000G	00	D4	002DB	CLRL	EDT\$G_INPUT_OPN	:	0991
00000000G	00	00	9E	002E1	MOVAB	EDT\$Z_EOB_LN, EDT\$A_WK_LN	:	0994
	50	68	D0	002EC	MOVL	EDT\$A_CUR_BUF, R0	:	0995
		0C	A0	B4 002EF	CLRW	12(R0)	:	
		2A	A0	94 002F2	CLRB	42(R0)	:	0996
		1C	AE	9F 002F5	PUSHAB	FILE_DESC	:	1003
00000000G	00	01	FB	002F8	CALLS	#1, STR\$FREE1_DX	:	
		24	AE	9F 002FF	PUSHAB	RHB_DESC	:	1004
00000000G	00	01	FB	00302	CALLS	#1, STR\$FREE1_DX	:	
		50	D4	00309	CLRL	R0	:	1010
		04	0030B	RET			:	1011

; Routine Size: 780 bytes, Routine Base: _EDT\$CODE + 0000

; 413 1012 1
 ; 414 1013 1 !<BLF/PAGE>

EDT\$WFREAINP
V04-000

EDT\$WFREAINP - read a line from the input file
EDT\$SRD_ILN - read a line from the input file

K 13
16-Sep-1984 02:11:53
14-Sep-1984 12:25:41

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

Page 13
(4)

: 416
: 417
: 418
1014 1 END
1015 1
1016 0 ELUDOM

! of module EDT\$WFREAINP

PSECT SUMMARY

Name Bytes Attributes
_EDT\$CODE 780 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	47	12	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\$:WFREAINP/OBJ=OBJ\$:WFREAINP MSRC\$:WFREAINP.BLI/UPDATE=(ENH\$:WFREAINP)

: Size: 780 code + 0 data bytes
: Run Time: 00:38.8
: Elapsed Time: 00:49.5
: Lines/CPU Min: 1570
: Lexemes/CPU-Min: 9347
: Memory Used: 253 pages
: Compilation Complete

0141 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

