

EEEEEEEEE	DDDDDDDDDD	TTTTTTTTTT
EEEEEEEEE	DDDDDDDDDD	TTTTTTTTTT
EEEEEEEEE	DDDDDDDDDD	TTTTTTTTTT
EEE	DDD	TTT
EEEEEEEEE	DDDDDDDDDD	TTT
EEEEEEEEE	DDDDDDDDDD	TTT
EEEEEEEEE	DDDDDDDDDD	TTT

FILEID**MAIN

N 5

MM MM AAAAAA IIIII NN NN
MM MM AAAAAA IIIII NN NN
MMMM Mmmm AA AA II NN NN
MMMM Mmmm AA AA II NN NN
MM MM MM AA AA III NNNN NN
MM MM MM AA AA III NNNN NN
MM MM AA AA AA III NN NN
MM MM AA AA AA III NN NN
MM MM AAAAAAAAAA III NN NNNN
MM MM AAAAAAAAAA III NN NNNN
MM MM AA AA AA III NN NN
MM MM AA AA AA III NN NN

.....

LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL SS SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

ED1
VO4

```

0001 0 XTITLE 'EDT$MAIN - main entry point'
0002 0 MODULE EDT$MAIN (
0003 0           IDENT = 'V04-000'
0004 0           ) =
0005 1 BEGIN
0006 1 ****
0007 1 *
0008 1 *      COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 *      DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 *      ALL RIGHTS RESERVED.
0011 1 *
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 This module contains the main entry point for callable EDT.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: 6-AUG-1979
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 18-FEB-1981. This module was created by
0045 1 extracting routine EDT from module EDTCTR.
0046 1 1-002 - Regularize headers, change from IO_STS to EDT$SG IOFI_NFND ,
0047 1 and from SY_EXIT to EDT$SFATAL_IOERR . JBS 19-Feb-1981
0048 1 1-003 - Fix module name and file name. JBS 04-Mar-1981
0049 1 1-004 - Change entry point name from EDT to EDT$MAIN ENTRY . JBS 26-Mar-1981
0050 1 1-005 - Change E VERSION to EDT$ST VERSION. JBS 30-Mar-1981
0051 1 1-006 - Change EDT$ST VERSION to EDT$ST VER . JBS 30-Mar-1981
0052 1 1-007 - Correct a spelling error in EDT$OPEN_IFIDEF . JBS 02-Apr-1981
0053 1 1-008 - Use the new message codes. JBS 04-Aug-1981
0054 1 1-009 - Fix reading startup file. JBS 16-Aug-1981
0055 1 1-010 - Implement SET NOFN. TMV 11-Sep-1981
0056 1 1-011 - Correct edit history. JBS 01-Oct-1981
0057 1 1-012- make major changes to create sharable image. STS 17-Nov-1981

```

58 0058 1 ! 1-013 - Put L_LINE back TMV 7-Dec-81
59 0059 1 ! 1-014 - Fix up DSC\$ macros and add comments. STS 7-Dec-1981
60 0060 1 ! 1-015 - Change the defaulting of the journal filename. STS 17-Dec-1981
61 0061 1 ! 1-016 - Change all i/o calls for command file to use EDT\$FILEIO. STS 25-Dec-1981
62 0062 1 ! 1-017 - Change call to edt\$fileio to use callfio. STS 07-Jan-1982
63 0063 1 ! 1-018 - Convert open output file to use callfio. STS 13-Jan-1982
64 0064 1 ! 1-019 - Deallocate dynamic strings. STS 10-Feb-1982
65 0065 1 ! 1-020 - Take out unused rhb descriptor. STS 19-Feb-1982
66 0066 1 ! 1-021 - Pass filename to edt\$fiopn err. STS 26-Feb-1982
67 0067 1 ! 1-022 - Add callable literals. STS 08-Mar-1982
68 0068 1 ! 1-023 - Enable a handler. STS 09-Mar-1982
69 0069 1 ! 1-024 - Remove the version number to a require file. JBS 09-Mar-1982
70 0070 1 ! 1-025 - Check and set re-entry flag. STS 11-Mar-1982
71 0071 1 ! 1-026 - Change EDT\$SG CMD and EDT\$SG JOU to EDT\$SV OPTIONS. JBS 24-Mar-1982
72 0072 1 ! 1-027 - File I/O routines return status. JBS 26-Mar-1982
73 0073 1 ! 1-028 - Add return value of 2 from EDT\$INIT. JBS 26-Mar-1982
74 0074 1 ! 1-029 - Copy the formal strings, in case they have an unusual format.
75 0075 1 ! JBS 26-Mar-1982
76 0076 1 ! 1-030 - Fix a typo in edit 1-027. JBS 27-Mar-1982
77 0077 1 ! 1-031 - Set up command file open flag. STS 07-Apr-1982
78 0078 1 ! 1-032 - Give messages on CLOSE errors. JBS 12-Apr-1982
79 0079 1 ! 1-033 - Add chaining between command files using SET COMMAND. JBS 04-May-1982
80 0080 1 ! 1-034 - Avoid double error printouts on the PDP-11. JBS 28-May-1982
81 0081 1 ! 1-035 - Conditionalize out variable set up for 11's. STS 01-Jun-1982
82 0082 1 ! 1-036 - Remove L_LINE and L_CHANGE. JBS 03-Jun-1982
83 0083 1 ! 1-037 - Remove command file looping. JBS 03-Jun-1982
84 0084 1 ! 1-038 - Allow a new return code from EDT\$INIT. JBS 03-Jun-1982
85 0085 1 ! 1-039 - Revise the interface to command files. JBS 07-Jun-1982
86 0086 1 ! 1-040 - Be sure the command name is blank after the startup file is read. JBS 11-Jun-1982
87 0087 1 ! 1-041 - Pass the default file name in the RHB parameter. JBS 15-Jun-1982
88 0088 1 ! 1-042 - Default the passed parameters if none passed. STS 13-Jul-1982
89 0089 1 ! 1-043 - Don't allow defaulting of the main input file. STS 22-Jul-1982
90 0090 1 ! 1-044 - Don't print the initial warning messages if we will enter change
mode, but do leave it for change mode to show. JBS 23-Jul-1982
91 0091 1 ! 1-045 - Check the status returned by EDT\$EXE_CMD. JBS 28-Jul-1982
92 0092 1 ! 1-046 - Add XLATE routine. STS 13-Aug-1982
93 0093 1 ! 1-047 - Make use of the literals for the options word masks. STS 17-Aug-1982
94 0094 1 ! 1-048 - Change the interface to EDT\$SET_COMFNAM. JBS 23-Aug-1982
95 0095 1 ! 1-049 - Fix up references to EDTSM_bits. JBS 23-Aug-1982
96 0096 1 !--
97 0097 1 !--
98 0098 1 !--

```
100      0099 1 %SBTTL 'Declarations'  
101      0100 1  
102      0101 1 TABLE OF CONTENTS:  
103      0102 1  
104      0103 1  
105      0104 1 REQUIRE 'EDTSRC:TRAROUNAM';  
106      0543 ,  
107      0544 1 FORWARD ROUTINE  
108      0545 1   EDT$EDIT;  
109      0546 1  
110      0547 1  
111      0548 1 INCLUDE FILES:  
112      0549 1 !  
113      0550 1  
114      0551 1 REQUIRE 'EDTSRC:EDTREQ';  
115      0686 1  
L 116      0687 1 %IF %BLISS (BLISS32)  
117      0688 1 %THEN  
118      0689 1  
119      0690 1 REQUIRE 'EDTSRC:SYSSYM';  
120      0720 1  
121      0721 1 %FI  
122      0722 1  
123      0723 1  
124      0724 1 MACROS:  
125      0725 1  
126      0726 1 +  
127      0727 1 Macro for the default name of the journal file. This is defined as  
128      0728 1 a macro so we can use %CHARCOUNT to pass the length of the string.  
129      0729 1 -  
130      0730 1 <BLF/NOFORMAT>  
131      0731 1  
132      0732 1 MACRO  
133      0733 1   JOU_DEFLT = %STRING ('.JOU:') % ; ! File type of .JOU, latest version.  
134      0734 1  
135      0735 1 <BLF/FORMAT>  
136      0736 1  
137      0737 1 EQUATED SYMBOLS:  
138      0738 1  
139      0739 1  
140      0740 1 EXTERNAL LITERAL  
141      0741 1   EDT$M_NOCREATE,  
142      0742 1   EDT$M_NOJOURNAL,  
143      0743 1   EDT$M_COMMAND,  
144      0744 1   EDT$M_NOCOMMAND,  
145      0745 1   EDT$K_OPEN_IN_OUT,  
146      0746 1   EDT$K_OPEN_INPUT,  
147      0747 1   EDT$K_INPUT_FILE,  
148      0748 1   EDT$K_JOURNAL_FILE,  
149      0749 1   EDT$K_OPEN_OUTPUT_NOSEQ,  
150      0750 1   EDT$K_COMMAND_FILE,  
151      0751 1   EDT$K_CLOSE;  
152      0752 1  
153      0753 1  
154      0754 1 OWN STORAGE:  
155      0755 1  
156      0756 1   NONE
```

EDT\$MAIN
V04-000

EDT\$MAIN - main entry point
Declarations

: 157 0757 1 |
: 158 0758 1 | EXTERNAL REFERENCES:
: 159 0759 1 |
: 160 0760 1 | Inside the routine.
: 161 0761 1 | -

E 6
16-Sep-1984 01:07:45
14-Sep-1984 12:23:51
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]MAIN.BLI;1
Page 4 (2)

EDT
V04

```
: 163 0762 1 %SBTTL 'EDT$EDIT - Main entry point'  
164 0763 1  
165 0764 1 GLOBAL ROUTINE EDT$EDIT  
166 0765 1 ! Main entry point  
167 L 0766 1 %IF %BLISS (BLISS32)  
168 0767 1 %THEN  
169 0768 1 (INP_DESC,  
170 0769 1 OUT_DESC,  
171 0770 1 CMD_DESC,  
172 0771 1 JOU_DESC,  
173 0772 1 OPTIONS_WRD,  
174 0773 1 FILE_ROUT,  
175 0774 1 WORK_ROUT,  
176 0775 1 XLATE_ROUT  
177 0776 1 )  
178 0777 1 %FI  
179 0778 1 =  
180 0779 1  
181 0780 1  
182 0781 1 !++  
183 0782 1 FUNCTIONAL DESCRIPTION:  
184 0783 1  
185 0784 1 The main entry point for the editor.  
186 0785 1  
187 0786 1 FORMAL PARAMETERS:  
188 0787 1  
189 0788 1 +  
190 0789 1 passed variables are set up in global data on the 11's in order to  
191 0790 1 save stack space.  
192 0791 1 -  
193 0792 1 INP_DESC - input file descriptor  
194 0793 1 OUT_DESC - output file descriptor  
195 0794 1 CMD_DESC - command file descriptor  
196 0795 1 JOU_DESC - journal file descriptor  
197 0796 1 OPTIONS_WRD  
198 0797 1 bit 0 = /RECOVER  
199 0798 1 bit 1 = /COMMAND  
200 0799 1 bit 2 = /NOJOURNAL  
201 0800 1 bit 3 = /NOOUTPUT  
202 0801 1 bit 4 = /NOCOMMAND  
203 0802 1 bit 5 = /NOCREATE  
204 0803 1 FILE_ROUT - address of file i/o routine to use  
205 0804 1 WORK_ROUT - address of work file routine to use  
206 0805 1 XLATE_ROUT - address for translation routine  
207 0806 1  
208 0807 1 IMPLICIT INPUTS:  
209 0808 1  
210 0809 1  
211 0810 1  
212 0811 1  
213 0812 1  
214 0813 1  
215 0814 1  
216 0815 1  
217 0816 1  
218 0817 1  
219 0818 1  
EDTSSA_INP_NAM  
EDTSSA_OUT_NAM  
EDTSSA_CMD_NAM  
EDTSSA_COM_NAM  
EDTSSA_JOU_NAM  
EDTSSG_INP_NAMLEN
```

```

: 220      0819 1    EDT$SG_OUT_NAMLEN
: 221      0820 1    EDT$SG_CMD_NAMLEN
: 222      0821 1    EDT$SG_JOU_NAMLEN
: 223      0822 1    EDT$SG_LASTMSG
: 224      0823 1    EDT$SA_CMD_BUF
: 225      0824 1    EDT$SA_CMD_END
: 226      0825 1
: 227      0826 1    ROUTINE VALUE:
: 228      0827 1
: 229      0828 1    Normally 1 or EDT$EDITORABO
: 230      0829 1
: 231      0830 1    SIDE EFFECTS:
: 232      0831 1
: 233      0832 1    Performs an edit.
: 234      0833 1
: 235      0834 1    --
: 236      0835 1
: 237      0836 2    BEGIN
: 238      0837 2
L 0838 2 %IF %BLISS (BLISS32)
: 240      0839 2 %THEN
: 241      0840 2
: 242      0841 2    BUILTIN
: 243      0842 2    ACTUALCOUNT;
: 244      0843 2
: 245      0844 2    MAP
: 246      0845 2    FILE_ROUT : REF VECTOR [2],
: 247      0846 2    WORK_ROUT : REF VECTOR [2],
: 248      0847 2    XLATE_ROUT : REF VECTOR [2],
: 249      0848 2    CMD_DESC : REF BLOCK [8, BYTE],
: 250      0849 2    JOU_DESC : REF BLOCK [8, BYTE],
: 251      0850 2    OUT_DESC : REF BLOCK [8, BYTE],
: 252      0851 2    INP_DESC : REF BLOCK [8, BYTE],
: 253      0852 2    OPTIONS_WRD : REF BITVECTOR [32];
: 254      0853 2
: 255      0854 2 %FI
: 256      0855 2
: 257      0856 2    EXTERNAL ROUTINE
: 258      0857 2    EDT$SCALLFIO,
: 259      0858 2    EDT$SEXECMD,
: 260      0859 2    EDT$STERMIN_RCOVMOD : NOVALUE,
: 261      0860 2    EDT$INIT,
: 262      0861 2    EDT$STERMIN,
: 263      0862 2    EDT$SFIOERR : NOVALUE,
: 264      0863 2    EDT$STY_CURLN,
: 265      0864 2    EDT$SFMT_MSG : NOVALUE,
: 266      0865 2    EDT$SET_COMFNAM;
: 267      0866 2
L 0867 2 %IF %BLISS (BLISS32)
: 269      0868 2 %THEN
: 270      0869 2
: 271      0870 2    EXTERNAL ROUTINE
: 272      0871 2    EDT$XLATE,
: 273      0872 2    EDT$FILEIO,
: 274      0873 2    EDT$WORKIO,
: 275      0874 2    EDT$SHANDLER,
: 276      0875 2    STR$COPY_DX,
```

! Contains address of work i/o routine and environment

! sets up handler and calls file handling routine

! Execute commands

! Handle end of journal file during /RECOVER

! Do the initial processing

! Do the final processing

! Print a message about a file error, with file name

! Type out the current line

! Format and print a message

! Open and close command files

! for copying all kinds of strings

```

277 0876 2 STR$FREE1_DX; ! for dynamic descriptor deallocation
278 0877 2
279 0878 2
280 0879 2 %FI
281 0880 2
282 0881 2 EXTERNAL
283 0882 2 EDT$SG_REENTRY, ! flag signifying re-entry
284 0883 2 EDT$SA_COM_NAM, ! User-specified command file name
285 0884 2 EDT$SA_CMD_NAM, ! Command file name
286 0885 2 EDT$ST_CMD_NAM_LEN, ! Command file name length
287 0886 2 EDT$ST_CMD_NAM_DEF1 : VECTOR [, BYTE], ! First default command file name (counted ASCII string)
288 0887 2 EDT$ST_CMD_NAM_DEF2 : VECTOR [, BYTE], ! First default command file default name (counted ASCII string)
289 0888 2 EDT$ST_CMD_NAM_DEF3 : VECTOR [, BYTE], ! Second default command file name (counted ASCII string)
290 0889 2 EDT$ST_CMD_NAM_DEF4 : VECTOR [, BYTE], ! Second default command file default name (counted ASCII string)
291 0890 2
292 0891 2 EDT$SG_EXITD, ! Did we exit with the last command?
293 0892 2 EDT$SA_JOU_NAM, ! Name of journal file
294 0893 2 EDT$SG_JOU_NAMLEN, ! Length of journal file name
295 0894 2 EDT$SG_RCOV_MOD, ! do a recovery flag
296 0895 2 EDT$SG_EDIT_MOD, ! Current editing mode
297 0896 2 EDT$SG_EDIT_DFLMOD, ! Default editing mode
298 0897 2 EDT$SA_INP_NAM, ! input file name
299 0898 2 EDT$SG_INP_NAMLEN, ! Length of above name
300 0899 2 EDT$SG_FNF_MSGFLG, ! File not found message flag
301 0900 2 EDT$SA_OUT_NAM, ! Output file name
302 0901 2 EDT$SG_OUT_NAMLEN, ! Output file name length
303 0902 2 EDT$SG_IOFI_NFND, ! 1 = open failed due to file not found
304 0903 2 EDT$SV_OPTIONS : BITVECTOR [32]. ! Options bits
305 0904 2 EDT$SG_LASTMSG, ! Message for change mode to print
306 0905 2 EDT$SA_CMD_BUF, ! Pointer to current location in command buffer
307 0906 2 EDT$ST_CMD_BUF, ! Command buffer
308 0907 2 EDT$SA_CMD_END, ! Pointer to end of command buffer
309 0908 2 EDT$SG_CC_DONE: ! 1 = control C actually aborted something
310 0909 2
311 L 0910 2 %IF XBLISS (BLISS32)
312 0911 2 %THEN
313 0912 2
314 0913 2 EXTERNAL
315 0914 2 EDT$SA_FILE_ROUT, ! address of fileio routine to use
316 0915 2 EDT$SG_FILE_ENV, ! environment word to pass fileio routine
317 0916 2 EDT$SA_XLATE_ROUT, ! address of translation routine
318 0917 2 EDT$SG_XLATE_ENV, ! environment word to pass xlate
319 0918 2 EDT$SA_WORK_ROUT, ! address of work i/o routine
320 0919 2 EDT$SG_WORK_ENV; ! environment word to pass work routine
321 0920 2
322 0921 2 %FI
323 0922 2
324 P 0923 2 MESSAGES ((COMFILNEX, REENTRY, COMFILNOP, COMFILCLO, NOJNLFIL, INPFILNEX, NONSTDFIL, EDITORABO, ABOBYCC,
325 0924 2 (TRC__IGN));
326 0925 2
327 L 0926 2 %IF XBLISS (BLISS32)
328 0927 2 %THEN
329 0928 2
330 0929 2 ENABLE
331 0930 2 FDT$SHANDLER ();
332 0931 2
333 0932 2 %FI

```

```
334      0933 2
335      0934 2 LOCAL
336      0935 2   STATUS,
337      0936 2   FILE_DESC : BLOCK [8, BYTE],
338      0937 2   RHB_DESC : BLOCK [8, BYTE],
339      0938 2
340      L 0939 2 %IF #BLISS (BLISS32)
341      0940 2 %THEN
342      0941 2   L_INP_DESC : BLOCK [8, BYTE];
343      0942 2   L_OUT_DESC : BLOCK [8, BYTE];
344      0943 2   L_CMD_DESC : BLOCK [8, BYTE];
345      0944 2   L_JOU_DESC : BLOCK [8, BYTE];
346      0945 2 %FI
347      0946 2
348      0947 2 PRIMARY FILE CONDITION,
349      0948 2 STRING_LENGTH;
350      0949 2
351      0950 2 !+
352      0951 2 | Check the re-entry flag, so that the caller cannot
353      0952 2 | re-enter EDT. A code of 67 indicates the everything
354      0953 2 | o.k., a code of 68 means this process has already
355      0954 2 | entered EDT. -Return an error message
356      0955 2 |
357      0956 2
358      0957 2 IF (.EDT$$G_REENTRY EQ 67) THEN EDT$$G_REENTRY = 68 ELSE RETURN (EDTS_REENTRY);
359      0958 2
360      0959 2 EDTSSG_EXITD = 0;
361      0960 2 !
362      0961 2 | On VAX/VMS we have fields defined for descriptors.
363      0962 2 | Fill them in.
364      0963 2 |
365      0964 2
366      L 0965 2 %IF #BLISS (BLISS32)
367      0966 2 %THEN
368      0967 2   FILE_DESC [DSCSB_DTYPE] = DSC$K_DTYPE_T;
369      0968 2   FILE_DESC [DSCSB_CLASS] = DSC$K_CLASS_D;
370      0969 2   FILE_DESC [DSCSA_POINTER] = 0;
371      0970 2   FILE_DESC [DSCSW_LENGTH] = 0;
372      0971 2   RHB_DESC [DSCSB_DTYPE] = DSC$K_DTYPE_T;
373      0972 2   RHB_DESC [DSCSB_CLASS] = DSC$K_CLASS_D;
374      0973 2   RHB_DESC [DSCSA_POINTER] = 0;
375      0974 2   RHB_DESC [DSCSW_LENGTH] = 0;
376      0975 2
377      0976 2 IF (ACTUALCOUNT () GTR 5) AND (.FILE_ROUT NEQ 0)
378      0977 2 THEN
379      0978 2   BEGIN
380      0979 2     EDTSSA_FILE_ROUT = .FILE_ROUT [0];    ! get address for fileio routine passed
381      0980 2     EDTSSG_FILE_ENV = .FILE_ROUT [1];    ! and environment word
382      0981 2   END
383      0982 2 ELSE
384      0983 2   BEGIN
385      0984 2     EDTSSA_FILE_ROUT = EDT$FILEIO;
386      0985 2     EDTSSG_FILE_ENV = 0;
387      0986 2   END;
388      0987 2
389      0988 2 IF (ACTUALCOUNT () GTR 6) AND (.WORK_ROUT NEQ 0)
390      0989 2 THEN
```

```
391 0990 3      BEGIN
392 0991 3      EDTSSA_WORK_ROUT = .WORK_ROUT [0];
393 0992 3      EDTSSG_WORK_ENV = .FILE_ROUT [1];
394 0993 3      END
395 0994 2      ELSE
396 0995 2      BEGIN
397 0996 2      EDTSSA_WORK_ROUT = EDT$WORKIO;
398 0997 2      EDTSSG_WORK_ENV = 0;
399 0998 2      END;
400 0999 2
401 1000 3      IF (ACTUALCOUNT () GTR 7) AND (.XLATE_ROUT NEQ 0)
402 1001 3      THEN
403 1002 3      BEGIN
404 1003 3      EDTSSA_XLATE_ROUT = .XLATE_ROUT [0];
405 1004 3      EDTSSG_XLATE_ENV = .XLATE_ROUT [1];
406 1005 3      END
407 1006 2      ELSE
408 1007 3      BEGIN
409 1008 3      EDTSSA_XLATE_ROUT = EDT$XLATE;
410 1009 3      EDTSSG_WORK_ENV = 0;
411 1010 2      END;
412 1011 2
413 1012 2      !+
414 1013 2      !+ Get all the filenames and the name lengths out of the descriptor
415 1014 2      in which they were passed and into global variables. On VMS we must
416 1015 2      first copy the formal to a local string, to be sure that the length
417 1016 2      and address fields mean what we think they do.
418 1017 2      !-
419 1018 2      L_INP_DESC [DSC$W_LENGTH] = 0;
420 1019 2      L_INP_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
421 1020 2      L_INP_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
422 1021 2      L_INP_DESC [DSC$A_POINTER] = 0;
423 1022 2      STR$COPY DX (L_INP_DESC, .INP_DESC);
424 1023 2      EDTSSA_INP_NAM = .L_INP_DESC [DSC$A_POINTER];
425 1024 2      EDTSSG_INP_NAMLEN = .L_INP_DESC [DSC$W_LENGTH];
426 1025 2      L_OUT_DESC [DSC$W_LENGTH] = 0;
427 1026 2      L_OUT_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
428 1027 2      L_OUT_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
429 1028 2      L_OUT_DESC [DSC$A_POINTER] = 0;
430 1029 2
431 1030 2      IF (ACTUALCOUNT () GTR 1) AND (.OUT_DESC NEQ 0) THEN STR$COPY_DX (L_OUT_DESC, .OUT_DESC);
432 1031 2
433 1032 2      EDTSSA_OUT_NAM = .L_OUT_DESC [DSC$A_POINTER];
434 1033 2      EDTSSG_OUT_NAMLEN = .L_OUT_DESC [DSC$W_LENGTH];
435 1034 2      L_CMD_DESC [DSC$W_LENGTH] = 0;
436 1035 2      L_CMD_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
437 1036 2      L_CMD_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
438 1037 2      L_CMD_DESC [DSC$A_POINTER] = 0;
439 1038 2
440 1039 2      IF (ACTUALCOUNT () GTR 2) AND (.CMD_DESC NEQ 0) THEN STR$COPY_DX (L_CMD_DESC, .CMD_DESC);
441 1040 2
442 1041 2      EDTSSA_CMD_NAM = .L_CMD_DESC [DSC$A_POINTER];
443 1042 2      EDTSSG_CMD_NAMLEN = .L_CMD_DESC [DSC$W_LENGTH];
444 1043 2      L_JOU_DESC [DSC$W_LENGTH] = 0;
445 1044 2      L_JOU_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
446 1045 2      L_JOU_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
447 1046 2      L_JOU_DESC [DSC$A_POINTER] = 0;
```

```
448 1047 2
449 1048 2 IF (ACTUALCOUNT () GTR 3) AND (.JOU_DESC NEQ 0) THEN STR$COPY_DX (L_JOU_DESC, .JOU_DESC);
450 1049 2
451 1050 2 EDT$SA_JOU_NAM = .L_JOU_DESC [DSC$A_POINTER];
452 1051 2 EDT$SG_JOU_NAMLEN = .L_JOU_DESC [DSC$W_LENGTH];
453 1052 2
454 1053 2 |+ Copy the options word into global storage so the rest of EDT can test it.
455 1054 2 |-
456 1055 2
457 1056 3 IF (ACTUALCOUNT () GTR 4) AND (.OPTIONS_WRD NEQ 0)
458 1057 2 THEN
459 1058 2 EDT$SV_OPTIONS = ..OPTIONS_WRD
460 1059 2 ELSE
461 1060 2 EDT$SV_OPTIONS = 0;
462 1061 2
463 1062 2 XFI
464 1063 2
465 1064 2 |+
466 1065 2 |+ Do the initial processing.
467 1066 2 |-
468 1067 2 PRIMARY_FILE_CONDITION = EDT$INIT ();
469 1068 2 |+
470 1069 2 |+ If initialization failed, abort right now.
471 1070 2 |-
472 1071 2
473 1072 2 |+ If (.PRIMARY_FILE_CONDITION EQL 3) THEN RETURN (EDTS_EDITORABO);
474 1073 2
475 1074 2 |+
476 1075 2 |+ If /NOCREATE was specified then exit if file was not found
477 1076 2 |-
478 1077 2
479 1078 3 |+ IF ((.PRIMARY_FILE_CONDITION EQL 0) AND ((.EDT$SV_OPTIONS AND EDT$M_NOCREATE) NEQ 0))
480 1079 2 |-
481 1080 3 |+ THEN
482 1081 3 BEGIN
483 1082 3 |+ IF .EDT$SG_FNF_MSGFLG THEN EDT$SFMT_MSG (EDTS_INPFILNEX);
484 1083 3
485 L 1084 3 |+ IF %BLISS (BLISS32)
486 1085 3 |+ THEN
487 1086 3 |+ STR$FREE1_DX (FILE_DESC);
488 1087 3 |+ STR$FREE1_DX (RHB_DESC);
489 1088 3 |+ STR$FREE1_DX (L_INP_DESC);
490 1089 3 |+ STR$FREE1_DX (L_OUT_DESC);
491 1090 3 |+ STR$FREE1_DX (L_CMD_DESC);
492 1091 3 |+ STR$FREE1_DX (L_JOU_DESC);
493 1092 3 |+ XFI
494 1093 3 |+ EDT$SG_REENTRY = 67;           ! set up reentry flag so we can call EDT again
495 1094 3 |+ RETURN(1);
496 1095 3 |+ END;
497 1096 2 |-
498 1097 2
499 1098 2 |+
500 1099 2 |+ Open the journal file. If recovery was specified, open it so we can read
501 1100 2 |+ from it.
502 1101 2 |-
503 1102 3 |+ BEGIN
504 1103 3
```

```
505      1104 3      LOCAL
506      1105 3      RHB_LEN;
507      1106 3
508      1107 3      RHB_LEN = %CHARCOUNT (JOU_DEFLT);
509      1108 3      STRING_DESC (RHB_DESC, RHB_LEN, CHSPTR (UPLIT (JOU_DEFLT)));
510      1109 2      END;
511      1110 2
512      1111 2      IF .EDT$SG_RCOV_MOD
513      1112 2      THEN
514      1113 2      BEGIN
515      1114 3      STRING_DESC (FILE_DESC, EDT$SG_JOU_NAMLEN, .EDT$SA_JOU_NAM);
516      1115 3      STATUS = EDT$$CALEFIO (EDTSK_OPEN_IN_OUT, EDTSK_JOURNAL_FILE, FILE_DESC, RHB_DESC);
517      1116 3
518      1117 4      IF ( NOT .STATUS)
519      1118 3      THEN
520      1119 4      BEGIN
521      1120 4      EDT$SFIOPN_ERR (EDTS_NOJNLFIL, FILE_DESC);
522      1121 4      RETURN (EDTS_EDITORABO);
523      1122 3      END;
524      1123 3
525      1124 3      END
526      1125 2      ELSE
527      1126 2      :+ We are not in recovery mode; just open the journal file for write.
528      1127 2      :-+
529      1128 2
530      1129 2
531      1130 3      IF ((.EDT$SV_OPTIONS AND EDT$M_NOJOURNAL) EQ 0)           ! NOT /NOJOURNAL
532      1131 2      THEN
533      1132 3      BEGIN
534      1133 3      STRING_DESC (FILE_DESC, EDT$SG_JOU_NAMLEN, .EDT$SA_JOU_NAM);
535      1134 3      STATUS = EDT$$CALEFIO (EDTSK_OPEN_OUTPUT_NGSEQ, EDTSK_JOURNAL_FILE, FILE_DESC, RHB_DESC);
536      1135 2      END;
537      1136 2
538      1137 2      :+ Make sure the journal file was opened.
539      1138 2      :-+
540      1140 2
541      1141 3      IF (( NOT .STATUS) AND (.EDT$SV_OPTIONS AND EDT$M_NOJOURNAL) EQ 0)
542      1142 2      THEN
543      1143 3      BEGIN
544      1144 3      EDT$SFIOPN_ERR (EDTS_NOJNLFIL, FILE_DESC);
545      1145 3      RETURN (EDTS_EDITORABO);
546      1146 2      END;
547      1147 2
548      1148 2      :+ Execute the startup file, if one was provided. The startup file is always executed
549      1149 2      in line mode.
550      1150 2      :-+
551      1151 2
552      1152 2      EDT$SG_EDIT_MODE = LINE_MODE;
553      1153 2
554      1154 3      IF ((.EDT$SV_OPTIONS AND EDT$M_NOCOMMAND) EQ 0)
555      1155 2      THEN
556      1156 3      BEGIN
557      1157 3
558      1158 4      IF (.EDT$SG_CMD_NAMLEN EQ 0)
559      1159 3      THEN
560      1160 4      BEGIN
```

```
562 1161 4 |+
563 1162 4 | No command file name was provided, use the default. Note that there are
564 1163 4 | two default names; if the first is not found we try the second.
565 1164 4 |-
566 1165 4 STATUS = EDT$$SET COMFNAM ( !  
567 1166 4 CH$PTR (EDT$ST CMD_NAM_DEF1, 1), ! Name  
568 1167 4 CH$RCHAR (EDT$ST CMD_NAM_DEF1), ! Length of name  
569 1168 4 CH$PTR (EDT$ST CMD_NAM_DEF2, 1), ! Default name  
570 1169 4 CH$RCHAR (EDT$ST_CMD_NAM_DEF2) ! Default name length  
571 1170 4 );
572 1171 4
573 1172 5 IF ( NOT .STATUS) !
574 1173 4 THEN
575 1174 4 STATUS = EDT$$SET COMFNAM ( !
576 1175 4 CH$PTR (EDT$ST CMD_NAM_DEF3, 1), ! Name  
577 1176 4 CH$RCHAR (EDT$ST CMD_NAM_DEF3), ! Length of name  
578 1177 4 CH$PTR (EDT$ST CMD_NAM_DEF4, 1), ! Default name  
579 1178 4 CH$RCHAR (EDT$ST_CMD_NAM_DEF4) ! Default name length  
580 1179 4 );
581 1180 4
582 1181 4 END
583 1182 3 ELSE
584 1183 3 STATUS = EDT$$SET COMFNAM (.EDT$SA_CMD_NAM, .EDT$G_CMD_NAMLEN, CH$PTR (EDT$ST_CMD_NAM_DEF4, 1),
585 1184 3 CH$RCHAR (EDT$ST_CMD_NAM_DEF4));
586 1185 3
587 1186 3 IF .STATUS
588 1187 3 THEN
589 1188 4 BEGIN
590 1189 4
591 1190 5 IF (EDT$SEXECMD (INP_COMMAND) EQ 2)
592 1191 4 THEN
593 1192 5 BEGIN
594 1193 5 |+
595 1194 5 | The startup file was terminated by control C.
596 1195 5 |-
597 1196 5
598 1197 5 IF .EDT$G_CC_DONE THEN EDT$SFMT_MSG (EDTS_ABOBYCC) ELSE EDT$SFMT_MSG (EDTS_CTRC__IGN);
599 1198 5
600 1199 5 EDT$SA_CMD_BUF = EDT$ST_CMD_BUF;
601 1200 5 EDT$SA_CMD_END = .EDT$SA_CMD_BUF;
602 1201 5 CH$WCHAR (%C'!', .EDT$SA_CMD_END);
603 1202 5 |+
604 1203 5 | Make sure we stay in line mode so the message is visible.
605 1204 5 |-
606 1205 5 EDT$G_EDIT_DFLTMOD = LINE_MODE;
607 1206 4 END;
608 1207 4
609 1208 4 END
610 1209 3 ELSE
611 1210 3
612 1211 4 IF ((.EDT$SV_OPTIONS AND EDT$M_COMMAND) NEQ 0)
613 1212 3 THEN
614 1213 4 BEGIN
615 1214 4 STRING_DESC (FILE_DESC, EDT$G_CMD_NAMLEN, .EDT$SA_CMD_NAM);
616 1215 4 EDT$SFOPEN_ERR (
617 1216 4
618 1217 4 IF .EDT$G_IOFI_NFND THEN EDTS_COMFILNEX ELSE EDTS_COMFILNOP, FILE_DESC);
```

```
619      1218 4
620      1219 4
621      1220 3
622      1221 3
623      1222 2
624      1223 2
625      1224 2
626      1225 2
627      1226 2
628      1227 2
629      1228 2
630      1229 2
631      1230 2
632      1231 2
633      1232 2
634      1233 2
635      1234 2
636      1235 2
637      1236 2
638      1237 2
639      1238 2
640      1239 2
641      1240 2
642      1241 2
643      1242 2
644      1243 2
645      1244 2
646      1245 2
647      1246 2
648      1247 2
649      1248 2
650      1249 2
651      1250 2
652      1251 2
653      1252 2
654      1253 2
655      1254 2
656      1255 2
657      1256 2
658      1257 2
659      1258 2
660      1259 2
661      1260 2
662      1261 2
663      1262 2
664      1263 2
665      1264 2
666      1265 2
667      1266 2
668      1267 2
669      1268 2
670      1269 2
671      1270 2
672      1271 2
673      1272 2
674      1273 2
675      1274 2

        RETURN (EDT$_EDITORABO);
        END;

        END;

        /* Make sure the command file is closed, and the name is null.
        STATUS = EDT$$SET_COMFNAM (0, 0, 0, 0);      ! Close the command file
        IF ( NOT .STATUS) THEN RETURN (EDT$_EDITORABO);

        /* Use the default mode setting.
        EDT$$G_EDIT_MOD = .EDT$$G_EDIT_DFLTMOD;
        /* Give an error message if the primary file was not found or does not have standard format,
        unless we are about to enter change mode, in which case defer the message.

        CASE .PRIMARY_FILE_CONDITION FROM 0 TO 2 OF
          SET
            [0] :
              IF (.EDT$$G_EDIT_MOD EQ LINE_MODE)
                THEN
                  BEGIN
                    IF .EDT$$G_FNF_MSGFLG THEN EDT$$FMT_MSG (EDT$_INPFILNEX);
                  END
                ELSE
                  IF .EDT$$G_FNF_MSGFLG THEN EDT$$G_LASTMSG = EDT$_INPFILNEX;
            [1] :
              BEGIN
                0
              END;
            [2] :
              IF (.EDT$$G_EDIT_MOD EQ LINE_MODE)
                THEN
                  EDT$$FMT_MSG (EDT$_NONSTDFIL)
                ELSE
                  EDT$$G_LASTMSG = EDT$_NONSTDFIL;
            TES;

        /* If we are in recovery mode, execute the remainder of the journal file.
        (Part of the journal file may have been used to respond to SUBSTITUTE/QUERY
        commands in the startup file.)
```

; R
:
:

```
676      1275 2 !-
677      1276 2
678      1277 3     IF (.EDT$SG_RCOV_MOD)
679      1278 2     THEN
680      1279 3       BEGIN
681      1280 3
682      1281 4     IF (EDT$SEXECMD (INP_JOURNAL) EQL 2)
683      1282 3     THEN
684      1283 4       BEGIN
685      1284 4     !+ The recovery process was interrupted by a control C. Bail out.
686      1285 4     !-
687      1286 4
688      1287 4     IF .EDT$SG_CC_DONE THEN RETURN (EDT$_ABOBYCC) ELSE RETURN (EDT$_CTRC__IGN);
689      1288 4
690      1289 4     END;
691      1290 3
692      1291 3
693      1292 3     EDT$STERMIN_RCOVMOD ();
694      1293 2     END;
695      1294 2
696      1295 2     !+ If we are in line mode, then print the current line.
697      1296 2     !-
698      1297 2
699      1298 2
700      1299 3     IF (.EDT$SG_EDIT_MOD EQL LINE_MODE)
701      1300 2     THEN
702      1301 3       BEGIN
703      1302 3       EDT$STY_CURLN ();
704      1303 2     END;
705      1304 2
706      1305 2     !+ And execute the main stream of commands.
707      1306 2     !-
708      1307 2       EDT$SEXECMD (INP_TERM);
709      1308 2
710      1309 2     !+ Do the finalization processing.
711      1310 2     !-
712      1311 2       EDT$STERMIN ();
713      1312 2
714      1313 2     !+ On the VAX we need to deallocate the dynamic descriptors
715      1314 2     !-
716      1315 2
717      1316 2
718      L 1317 2     XIF %BLISS (BLISS32)
719      1318 2     XTHEN
720      1319 2       STR$FREE1_DX (FILE_DESC);
721      1320 2       STR$FREE1_DX (RMB_DESC);
722      1321 2       STR$FREE1_DX (L_INP_DESC);
723      1322 2       STR$FREE1_DX (L_OUT_DESC);
724      1323 2       STR$FREE1_DX (L_CMD_DESC);
725      1324 2       STR$FREE1_DX (L_JOU_DESC);
726      1325 2     XFI
727      1326 2
728      1327 2     EDT$SG_REENTRY = 67;           ! set up reentry flag so we can call EDT again
729      1328 2     RETURN(1);                  ! of routine EDT$EDIT
730      1329 1     END;
```

EDT\$MAIN
V04-000

EDT\$MAIN - main entry point
EDT\$EDIT - Main entry point

C 7
16-Sep-1984 01:07:45 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:23:51 DISK\$VMSMASTER:[EDT.SRC]MAIN.BLI;1

Page 15
(3)

.TITLE EDT\$MAIN EDT\$MAIN - main entry point
.IDENT \V04-000\

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

00 00 00 3B 55 4F 4A 2E 00000 P.AAA: .ASCII \.JOU;\<0>\<0>\<0>

.EXTRN EDT\$M_NOCREATE, EDT\$M_NOJOURNAL
.EXTRN EDT\$M_COMMAND, EDT\$M_NOCOMMAND
.EXTRN EDT\$K_OPEN_IN_OUT
.EXTRN EDT\$K_OPEN_INPUT
.EXTRN EDT\$K_INPUT_FILE
.EXTRN EDT\$K_JOURNAL_FILE
.EXTRN EDT\$K_OPEN_OUTPUT_NOSEQ
.EXTRN EDT\$K_COMMAND_FILE
.EXTRN EDT\$K_CLOSE, EDT\$SCALLFIO
.EXTRN EDT\$SEXECMD, EDT\$TERMIN_RCOVMOD
.EXTRN EDT\$INIT, EDT\$TERMIN
.EXTRN EDT\$FIOPNERR, EDT\$STY_CURLN
.EXTRN EDT\$SFMTMSG, EDT\$SET_COMFNAM
.EXTRN EDT\$XLATE, EDT\$FILEIO
.EXTRN EDT\$WORKIO, EDT\$HANDLER
.EXTRN STR\$COPY_DX, STR\$FREE1_DX
.EXTRN EDT\$SG_REENTRY, EDT\$SA_COM_NAM
.EXTRN EDT\$SA_CMD_NAM, EDT\$SG_CMD_NAMLEN
.EXTRN EDT\$ST_CMD_NAM_DEF1
.EXTRN EDT\$ST_CMD_NAM_DEF2
.EXTRN EDT\$ST_CMD_NAM_DEF3
.EXTRN EDT\$ST_CMD_NAM_DEF4
.EXTRN EDT\$SG_EXITD, EDT\$SA_JOU_NAM
.EXTRN EDT\$SG_JOU_NAMLEN
.EXTRN EDT\$SG_RCOV_MOD
.EXTRN EDT\$SG_EDIT_MOD
.EXTRN EDT\$SG_EDIT_DFLTMOD
.EXTRN EDT\$SA_INP_NAM, EDT\$SG_INP_NAMLEN
.EXTRN EDT\$SG_FNF_MSGFLG
.EXTRN EDT\$SA_OUT_NAM, EDT\$SG_OUT_NAMLEN
.EXTRN EDT\$SG_IOFI_NFND
.EXTRN EDT\$SV_OPTIONS, EDT\$SG_LASTMSG
.EXTRN EDT\$SA_CMD_BUF, EDT\$ST_CMD_BUF
.EXTRN EDT\$SA_CMD_END, EDT\$SG_CC_DONE
.EXTRN EDT\$SA_FILE_ROUT
.EXTRN EDT\$SG_FILE_ENV
.EXTRN EDT\$SA_XLATE_ROUT
.EXTRN EDT\$SG_XLATE_ENV
.EXTRN EDT\$SA_WORK_ROUT
.EXTRN EDT\$SG_WORK_ENV
.EXTRN EDT\$COMFILNEX, EDT\$REENTRY
.EXTRN EDT\$COMFILNOP, EDT\$COMFILCLO
.EXTRN EDT\$NOJNLFIL, EDT\$INPFILNEX
.EXTRN EDT\$NONSTDFIL, EDT\$EDITORABO
.EXTRN EDT\$ABOBYCC, EDT\$CTR_C__IGN
.EXTRN STR\$COPY_R

OFFC 00000
5B 00000000G 00 9E 00002

.ENTRY EDT\$EDIT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-: 0767
R11
MOVAB EDT\$SA_CMD_NAM, R11

		5A 00000000G	00	9E 00009	MOVAB	EDTSSG_WORK_ENV, R10	
		59 00000000G	00	9E 00010	MOVAB	EDTSS\$CMD_NAM_DEF4, R9	
		58 00000000G	00	9E 00017	MOVAB	STRSCOPY_R-, R8	
		57 00000000G	00	9E 0001E	MOVAB	STRSCOPY-DX, R7	
		56 00000000G	00	9E 00025	MOVAB	EDTSSG_EDIT_MOD, R6	
		55 00000000G	00	9E 0002C	MOVAB	STR\$FREE1_DX, R5	
		54 00000000G	00	9E 00033	MOVAB	EDTSSV_OPTIONS, R4	
		5E	34	C2 0003A	SUBL2	#52, SP	
	00000043	6D 042E	CF	DE 0003D	MOVAL	45S (FP)	0836
		8F 00000000G	00	D1 00042	CMPL	EDTSSG_REENTRY, #67	0957
		00000000G	00	44	BNEQ	1S	
		8F 9A 0004F	0A	12 0004D	MOVZBL	#68, EDTSSG_REENTRY	
		50 00000000G	08	11 00057	BRB	2S	
		8F D0 00059	1S:	04 00060	MOVL	#EDTS_REENTRY, R0	
					RET		
		00000000G	J0	D4 00061	CLRL	EDTSSG_EXITD	0959
	2C	AE 020E0000	8F	D0 00067	MOVL	#34471936, FILE_DESC	0970
		30 AE 0006F	AE	D4 0006F	CLRL	FILE_DESC+4	0969
	24	AE 020E0000	8F	D0 00072	MOVL	#34471936, RHB_DESC	0974
		28 AE 0007A	AE	D4 0007A	CLRL	RHB_DESC+4	0973
		05	6C 91 0007D	CMPB	(AP), #5		0976
		1A 1B 00080	BLEQU	3S			
		18 AC 00082	TSTL	FILE_ROUT			
		15 13 00085	BEQL	3S			
	00000000G	50 18 AC 00087	MOVL	FILE_ROUT, R0		0979	
	00000000G	00 60 D0 0008B	MOVL	(R0), EDTSSA_FILE_ROUT			
	00000000G	00 04 A0 D0 00092	MOVL	4(R0), EDTSSG_FILE_ENV		0980	
	00000000G	11 11 0009A	BRB	4S		0976	
	00000000G	00 9E 0009C	MOVAB	EDTSFILEIO, EDTSSA_FILE_ROUT		0984	
		00000000G 00 D4 000A7	CLRL	EDTSSG_FILE_ENV		0985	
		06 6C 91 000AD	CMPB	(AP), #6		0988	
		17 1B 000B0	BLEQU	5S			
		1C AC 000B2	TSTL	WORK_ROUT			
		12 13 000B5	BEQL	5S			
	00000000G	00 1C BC D0 000B7	MOVL	WORK_ROUT, EDTSSA_WORK_ROUT		0991	
		50 18 AC D0 000BF	MOVL	FILE_ROUT, R0		0992	
		6A 04 A0 D0 000C3	MOVL	4(R0), EDTSSG_WORK_ENV			
	00000000G	00 0D 11 000C7	BRB	6S		0988	
	00000000G	00 9E 000C9	MOVAB	EDTSWORKIO, EDTSSA_WORK_ROUT		0996	
		00 D4 000D4	CLRL	EDTSSG_WORK_ENV		0997	
		07 6C 91 000D6	CMPB	(AP), #7		1000	
		1A 1B 000D9	BLEQU	7S			
		20 AC D5 000DB	TSTL	XLATE_ROUT			
		15 13 000DE	BEQL	7S			
	00000000G	50 20 AC D0 000E0	MOVL	XLATE_ROUT, R0		1003	
	00000000G	00 60 D0 000E4	MOVL	(R0), EDTSSA_XLATE_ROUT			
	00000000G	00 04 A0 D0 000EB	MOVL	4(R0), EDTSSG_XLATE_ENV		1004	
	00000000G	00 0D 11 000F3	BRB	8S		1000	
	00000000G	00 9E 000F5	MOVAB	EDTSXLATE, EDTSSA_XLATE_ROUT		1008	
		00 6A D4 00100	CLRL	EDTSSG_WORK_ENV		1009	
	1C AE 020E0000	8F D0 00102	MOVL	#34471936, L_INP_DESC		1018	
		20 AE D4 0010A	CLRL	L_INP_DESC+4		1021	
		04 AC DD 0010D	PUSHL	INP_DESC		1022	
		20 AE 9F 00110	PUSHAB	L_INP_DESC			
	00000000G	67 02 FB 00113	CALLS	#2, STR\$COPY_DX			
	00000000G	00 20 AE D0 00116	MOVL	L_INP_DESC+4, EDTSSA_INP_NAM		1023	
	00000000G	00 1C AE 3C 0011E	MOVZWL	L_INP_DESC, EDTSSG_INP_NAMLEN		1024	

14	AE 020E0000	8F	D0 00126	MOVL	#34471936, L_OUT_DESC	1025
01		18	AE D4 0012E	CLRL	L_OUT_DESC+4	1028
		6C	91 00131	CMPB	(AP), #1	1030
		0E	1B 00134	BLEQU	9\$	
		08	AC D5 00136	TSTL	OUT_DESC	
		09	13 00139	BEQL	9\$	
		08	AC DD 0013B	PUSHL	OUT_DESC	
		18	AE 9F 0013E	PUSHAB	L_OUT_DESC	
00000000G	67 00	02	FB 00141	CALLS	#2, STR\$COPY_DX	
00000000G	00	18	AE D0 00144	MOVL	L_OUT_DESC+4, EDTSSA_OUT_NAM	1032
00000000G	00	14	AE 3C 0014C	MOVZWL	L_OUT_DESC, EDTSSG_OUT_NAMLEN	1033
OC	AE 020E0000	8F	D0 00154	MOVL	#3447T936, L_CMD_DESC	1034
		10	AE D4 0015C	CLRL	L_CMD_DESC+4	1037
		02	6C 91 0015F	CMPB	(AP), #2	1039
		0E	1B 00162	BLEQU	10\$	
		0C	AC D5 00164	TSTL	CMD_DESC	
		09	13 00167	BEQL	10\$	
		0C	AC DD 00169	PUSHL	CMD_DESC	
		10	AE 9F 0016C	PUSHAB	L_CMD_DESC	
00000000G	67 6B 00	02	FB 0016F	CALLS	#2, STR\$COPY_DX	
00000000G	00	10	AE D0 00172	MOVL	L_CMD_DESC+4, EDTSSA_CMD_NAM	1041
04	AE 020E0000	0C	AE 3C 00176	MOVZWL	L_CMD_DESC, EDTSSG_CMD_NAMLEN	1042
		8F	D0 0017E	MOVL	#3447T936, L_JOU_DESC	1043
		08	AE D4 00186	CLRL	L_JOU_DESC+4	1046
		03	6C 91 00189	CMPB	(AP), #3	1048
		0E	1B 0018C	BLEQU	11\$	
		10	AC D5 0018E	TSTL	JOU_DESC	
		09	13 00191	BEQL	11\$	
		10	AC DD 00193	PUSHL	JOU_DESC	
		08	AE 9F 00196	PUSHAB	L_JOU_DESC	
00000000G	67 00	02	FB 00199	CALLS	#2, STR\$COPY_DX	
00000000G	00	08	AE D0 0019C	MOVL	L_JOU_DESC+4, EDTSSA_JOU_NAM	1050
04		04	AE 3C 001A4	MOVZWL	L_JOU_DESC, EDTSSG_JOU_NAMLEN	1051
		6C	91 001AC	CMPB	(AP), #4	1056
		0B	1B 001AF	BLEQU	12\$	
		14	AC D5 001B1	TSTL	OPTIONS_WRD	
		06	13 001B4	BEQL	12\$	
64	14	BC	D0 001B6	MOVL	@OPTIONS_WRD, EDTSSV_OPTIONS	1058
		02	11 001BA	BRB	13\$	
00000000G	00	64	D4 001BC	CLRL	EDTSSV_OPTIONS	
		00	FB 001BE	CALLS	#0, EDTSSINIT	1060
53		50	DO 001C5	MOVL	RO_PRIMARY_FILE_CONDITION	1067
03		53	D1 001C8	CMPL	PRIMARY_FILE_CONDITION, #3	1072
		03	12 001CB	BNEQ	14\$	
		01C1	31 001CD	BRW	31\$	
		53	D5 001D0	TSTL	PRIMARY_FILE_CONDITION	1078
		20	12 001D2	BNEQ	16\$	
00000000G	8F	64	D3 001D4	BITL	EDTSSV_OPTIONS, #EDTSM_NOCREATE	
		17	13 001DB	BEQL	16\$	
OD 00000000G	00	E9	001DD	BLAC	EDTSSG_FNF_MSGFLG, 15\$	
00000000G	00	8F	DD 001E4	PUSHL	#EDTS_INPFILNEX	1082
		01	FB 001EA	CALLS	#1, EDTSSFMT_MSG	
		024B	31 001F1	BRW	44\$	
6E		05	D0 001F4	MOVL	#5, RHB_LEN	1086
	FDFD	CF	9F 001F7	PUSHAB	RHB_LEN	1107
	04	AE	9F 001FB	PUSHAB	RHB_DESC	1108
	2C	AE	9F 001FE	PUSHAB		

68	33	00000000G	03	FB	00201	CALLS	#3, STRSCOPY_R		1111
		00000000G	00	E9	00204	BLBC	EDTSSG_RCOV_MOD, 17\$		1114
		00000000G	00	DD	0020B	PUSHL	EDTSSA_JOU_NAM		
			00	9F	00211	PUSHAB	EDTSSG_JOU_NAMLEN		
			34	AE	9F	PUSHAB	FILE_DESC		
68			03	FB	0021A	CALLS	#3, STRSCOPY_R		1115
			24	AE	9F	PUSHAB	RHB_DESC		
			30	AE	9F	PUSHAB	FILE_DESC		
		00000000G	8F	DD	00223	PUSHL	#EDT\$K_JOURNAL_FILE		
		00000000G	8F	DD	00229	PUSHL	#EDT\$K_OPEN_IN_OUT		
00000000G	00		04	FB	0022F	CALLS	#4, EDTSSCALFIO		1117
	52		50	DO	00236	MOVL	RO, STATUS		1120
	51		52	E8	00239	BLBS	STATUS, 20\$		1130
00000000G	8F		43	11	0023C	BRB	19\$		
			64	D3	0023E	17\$:	EDTSSV_OPTIONS, #EDT\$M_NOJOURNAL		
			2E	12	00245	BNEQ	18\$		
		00000000G	00	DD	00247	PUSHL	EDTSSA_JOU_NAM		1133
		00000000G	00	9F	0024D	PUSHAB	EDTSSG_JOU_NAMLEN		
			34	AE	9F	PUSHAB	FILE_DESC		
68			03	FB	00256	CALLS	#3, STRSCOPY_R		1134
			24	AE	9F	PUSHAB	RHB_DESC		
			30	AE	9F	PUSHAB	FILE_DESC		
		00000000G	8F	DD	0025F	PUSHL	#EDT\$K_JOURNAL_FILE		
		00000000G	8F	DD	00265	PUSHL	#EDT\$K_OPEN_OUTPUT_NOSEQ		
00000000G	00		04	FB	0026B	CALLS	#4, EDTSSCALFIO		1141
	52		50	DO	00272	MOVL	RO, STATUS		
	15		52	E8	00275	18\$:	BLBS	STATUS, 20\$	
00000000G	8F		64	D3	00278	BITL	EDTSSV_OPTIONS, #EDT\$M_NOJOURNAL		
			0C	12	0027F	BNEQ	20\$		
		00000000G	2C	AE	9F	00281	19\$:	FILE_DESC	1144
		00000000G	8F	DD	00284	PUSHL	#EDT\$NOJNLFIL		
			00EA	31	0028A	BRW	29\$		
00000000G	66		01	DO	0028D	20\$:	MOVL	#1, EDTSSG_EDIT_MOD	1152
	8F		64	D3	00290	BITL	EDTSSV_OPTIONS, #EDT\$M_NOCOMMAND		1154
			68	12	00297	BNEQ	24\$		
		50 00000000G	00	DO	00299	MOVL	EDTSSG_CMD_NAMLEN, RO		1158
			3C	12	002A0	BNEQ	21\$		
	7E 00000000G		00	9A	002A2	MOVZBL	EDT\$ST_CMD_NAM_DEF2, -(SP)		1169
		00000000G	00	9F	002A9	PUSHAB	EDT\$ST_CMD_NAM_DEF2+1		1168
	7E 00000000G		00	9A	002AF	MOVZBL	EDT\$ST_CMD_NAM_DEF1, -(SP)		1169
		00000000G	00	9F	002B6	PUSHAB	EDT\$ST_CMD_NAM_DEF1+1		1166
00000000G	00		04	FB	002BC	CALLS	#4, EDTSSSET_COMFNAM		1169
	52		50	DO	002C3	MOVL	RO, STATUS		
	2C		52	E8	002C6	BLBS	STATUS, 23\$		1172
	7E		69	9A	002C9	MOVZBL	EDT\$ST_CMD_NAM_DEF4, -(SP)		1178
		01	A9	9F	002CC	PUSHAB	EDT\$ST_CMD_NAM_DEF4+1		1177
	7E 00000000G		00	9A	002CF	MOVZBL	EDT\$ST_CMD_NAM_DEF3, -(SP)		1178
		00000000G	00	9F	002D6	PUSHAB	EDT\$ST_CMD_NAM_DEF3+1		1175
			0A	11	002DC	BRB	22\$		1178
	7E		69	9A	002DE	21\$:	MOVZBL	EDT\$ST_CMD_NAM_DEF4, -(SP)	1184
		01	A9	9F	002E1	PUSHAB	EDT\$ST_CMD_NAM_DEF4+1		1183
			50	DD	002E4	PUSHL	RO		1184
00000000G	00		6B	DD	002E6	PUSHL	EDTSSA_CMD_NAM		
	52		04	FB	002E8	22\$:	CALLS	#4, EDTSSSET_COMFNAM	
	53		50	DO	002EF	MOVL	RO, STATUS		
			52	E9	002F2	BLBC	STATUS, 27\$		1186
			02	DD	002F5	PUSHL	#2		1190

EDT\$MAIN
V04-000

EDTSMMAIN - main entry point
EDTSEEDIT - Main entry point

G 7
16-Sep-1984 01:07:45 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:23:51 DISK\$VMSMASTER:[EDT.SRC]MAIN.BLI;1

Page 19
(3)

EDT
V04

00000000G	00	01	FB	002F7		CALLS	#1, EDT\$SEXECMD	
	02	50	D1	002FE	24\$:	CMPL	R0, #2	
	08	7D	12	00301		BNEQ	30\$	
00000000G	00	E9	00303			BLBC	EDT\$SG_CC_DONE, 25\$	
00000000G	8F	DD	0030A			PUSHL	#EDTS_ABODYCC	
	06	11	00310			BRB	26\$	
00000000G	00	8F	DD	00312	25\$:	PUSHL	#EDTS_CTRCIGN	
00000000G	00	01	FB	00318	26\$:	CALLS	#1, EDT\$SFMTMSG	
00000000G	00	9E	0031F			MOVAB	EDT\$ST_CMD_BUF, EDT\$SA_CMD_BUF	1199
00000000G	00	D0	0032A			MOVL	EDT\$SA_CMD_BUF, EDT\$SA_CMD_END	1200
50 00000000G	00	D0	00335			MOVL	EDT\$SA_CMD_END, R0	1201
	60	21	90	0033C		MOVB	#33, (R0)	
00000000G	00	01	D0	0033F		MOVL	#1, EDT\$G_EDIT_DFLTMOD	1205
	38	11	00346			BRB	30\$	1186
00000000G	8F	64	D3	00348	27\$:	BITL	EDT\$SV_OPTIONS, #EDTS_COMMAND	1211
		2F	13	0034F		BEQL	30\$	
		68	DD	00351		PUSHL	EDT\$SA_CMD_NAM	1214
00000000G	00	9F	00353			PUSHAB	EDT\$SG_CMD_NAMLEN	
	34	AE	9F	00359		PUSHAB	FILE DESC	
	68	03	FB	0035C		CALLS	#3, STRSCOPY_R	
	2C	AE	9F	0035F		PUSHAB	FILE DESC	1215
08 00000000G	00	E9	00362			BLBC	EDT\$G_IIFI_NFND, 28\$	1217
00000000G	8F	DD	00369			PUSHL	#EDTS_COMFI[NEX	
	06	11	0036F			BRB	29\$	
00000000G	00	8F	DD	00371	28\$:	PUSHL	#EDTS_COMFILNOP	
	02	FB	00377	29\$:		CALLS	#2, EDT\$FIOPN_ERR	
	11	11	0037E			BRB	31\$	1219
	7E	7C	00380	30\$:		CLRQ	-(SP)	1227
	7E	7C	00382			CLRQ	-(SP)	
00000000G	00	04	FB	00384		CALLS	#4, EDT\$SET_COMFNAM	
	52	50	D0	0038B		MOVL	R0, STATUS	
	08	52	E8	0038E		BLBS	STATUS, 32\$	1229
50 00000000G	8F	D0	00391	31\$:		MOVL	#EDTS_EDITORABO, R0	
	04	00398				RET		
02 002D	66 00000000G	00	D0	00399	32\$:	MOVL	EDT\$G_EDIT_DFLTMOD, EDT\$G_EDIT_MOD	1234
	00	53	CF	003A0	33\$:	CASEL	PRIMARY_FILE_CONDITION, #0, #2	1240
	004C	0006	003A4	33\$:		.WORD	34\$-33\$,- 39\$-33\$,- 36\$-33\$	
	50 00000000G	00	D0	003AA	34\$:	MOVL	EDT\$G_FNF_MSGFLG, R0	1249
	01	66	D1	003B1		CMPL	EDT\$G_EDIT_MOD, #1	1245
		0B	12	003B4		BNEQ	35\$	
	37	50	E9	003B6		BLBC	R0, 39\$	1249
	00000000G	8F	DD	003B9		PUSHL	#EDTS_INPFILNEX	
		1B	11	003BF		BRB	37\$	
00000000G	00 00000000G	50	E9	003C1	35\$:	BLBC	R0, 39\$	1254
		8F	D0	003C4		MOVL	#EDTS_INPFILNEX, EDT\$G_LASTMSG	1245
		1F	11	003CF		BRB	39\$	
	01	66	D1	003D1	36\$:	CMPL	EDT\$G_EDIT_MOD, #1	1263
		0F	12	003D4		BNEQ	38\$	
00000000G	00 00000000G	8F	DD	003D6	37\$:	PUSHL	#EDTS_NONSTDFFIL	1265
		01	FB	003DC	37\$:	CALLS	#1, EDT\$SFMTMSG	
		0B	11	003E3		BRB	39\$	
00000000G	00 00000000G	8F	D0	003E5	38\$:	MOVL	#EDTS_NONSTDFFIL, EDT\$G_LASTMSG	1267
	2C 00000000G	00	E9	003F0	39\$:	BLBC	EDT\$G_RCOV_MOD, 42\$	1277
		03	DD	003F7		PUSHL	#3	1281
00000000G	00	01	FB	003F9		CALLS	#1, EDT\$SEXECMD	

EDT\$MAIN
V04-000EDT\$MAIN - main entry point
EDT\$EDIT - Main entry pointM 7
16-Sep-1984 01:07:45
14-Sep-1984 12:23:51
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]MAIN.BLI;1Page 20
(3)EDT
V04

02		50	D1	00400	CMPL	R0	#2		
08 00000000G	00	17	12	00403	BNEQ	41\$			
50 00000000G	8F	00	E9	00405	BLBC	EDT\$SG_CC_DONE,	40\$	1288	
		8F	D0	0040C	MOVL	#EDT\$_ABOBYCC,	R0		
		04	00413		RET				
50 00000000G	8F	D0	00414	40\$:	MOVL	#EDT\$_CTRC__IGN.	R0		
		04	0041B		RET				
00000000G	00	00	FB	0041C	41\$:	CALLS	#0, EDT\$STERMIN_RCOVMOD	1292	
01		66	D1	00423	42\$:	CMPL	EDT\$SG_EDIT_MOD, #1	1299	
00000000G	00	07	12	00426	BNEQ	43\$			
		00	FB	00428	CALLS	#0, EDT\$STY_CURLN		1302	
00000000G	00	7E	D4	0042F	43\$:	CLRL	-(SP)	1308	
00000000G	00	01	FB	00431	CALLS	#1, EDT\$SEXEC_CMD			
00000000G	00	00	FB	00438	CALLS	#0, EDT\$STERMIN		1312	
		2C	AE	9F	44\$:	PUSHAB	FILE_DESC	1319	
65		01	FB	00442	CALLS	#1, STR\$FREE1_DX			
65		24	AE	9F	00445	PUSHAB	RHB_DESC	1320	
65		01	FB	00448	CALLS	#1, STR\$FREE1_DX			
65		1C	AE	9F	0044B	PUSHAB	L_INP_DESC	1321	
65		01	FB	0044E	CALLS	#T, STR\$FREE1_DX			
65		14	AE	9F	00451	PUSHAB	L_OUT_DESC	1322	
65		01	FB	00454	CALLS	#T, STR\$FREE1_DX			
65		0C	AE	9F	00457	PUSHAB	L_CMD_DESC	1323	
65		01	FB	0045A	CALLS	#T, STR\$FREE1_DX			
65		04	AE	9F	0045D	PUSHAB	L_JOU_DESC	1324	
00000000G	00	65	01	FB	00460	CALLS	#T, STR\$FREE1_DX		
		43	8F	9A	00463	MOVZBL	#67, EDT\$SG_REENTRY	1327	
		50	01	D0	0046B	MOVL	#1, R0	1328	
			04	0046E	RET			1329	
			0000	0046F	45\$:	.WORD	Save nothing	0836	
			7E	D4	00471	CLRL	-(SP)		
			5E	DD	00473	PUSHL	SP		
00000000G	00	04	AC	7D	00475	MOVQ	4(AP), -(SP)		
		03	FB	00479	CALLS	#3, EDT\$SHANDLER			
		04	00480		RET				

: Routine Size: 1153 bytes, Routine Base: _EDT\$CODE + 0008

: 731 1330 1
: 732 1331 1 :<BLF/PAGE>S
R
E
L
M
C

EDT\$MAIN
V04-000

EDT\$MAIN - main entry point
EDT\$EDIT - Main entry point

16-Sep-1984 01:07:45 17
14-Sep-1984 12:23:51 VAX-11 Bliss-32 v4.0-742
DISK\$VMSMASTER:[EDT.SRC]MAIN.BLI;1

Page 21
(4)

: 734 1332 1 END
: 735 1333 1
: 736 1334 0 ELUDOM

! of module EDT\$MAIN

**F

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	1161	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols	Pages	Processing
	Loaded	Percent	Mapped	Time
\$255\$DUA2B:[EDT.SRC]EDT.L32;1	377	6	1	00:00.2
\$255\$DUA2B:[EDT.SRC]PSECTS.L32;1	2	1	50	00:00.1
\$255\$DUA2B:[SYSLIB]STARLET.L32;1	9776	6	0	00:04.2
			581	

COMMAND QUALIFIERS

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

: Size: 1153 code + 8 data bytes
: Run Time: 00:50.2
: Elapsed Time: 01:05.5
: Lines/CPU Min: 1593
: Lexemes/CPU-Min: 6856
: Memory Used: 284 pages
: Compilation Complete

0137 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

