


```

MM      MM      AAAAAA      IIIIII      NN      NN
MM      MM      AAAAAA      IIIIII      NN      NN
MMMM    MMMM    AA          AA      II      NN      NN
MMMM    MMMM    AA          AA      II      NN      NN
MM      MM      MM      AA      AA      II      NNNN     NN
MM      MM      MM      AA      AA      II      NNNN     NN
MM      MM      MM      AA      AA      II      NN      NN      NN
MM      MM      MM      AA      AA      II      NN      NN      NN
MM      MM      MM      AAAAAAAAAA      II      NN      NNNN
MM      MM      MM      AAAAAAAAAA      II      NN      NNNN
MM      MM      AA          AA      II      NN      NN
MM      MM      AA          AA      II      NN      NN
MM      MM      AA          AA      IIIIII     NN      NN
MM      MM      AA          AA      IIIIII     NN      NN

```

```

....
....
....
....

```

```

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$MAIN - main entry point'
2 0002 0 MODULE EDT$MAIN ( ! Main entry point for callable EDT
3 0003 0 IDENT = 'V04-000' ! File: MAIN.BLI Edit: JBS1049
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 contains the main entry point for callable EDT.
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: 6-AUG-1979
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 18-FEB-1981. This module was created by
45 0045 1 extracting routine EDT from modul: EDTCTR.
46 0046 1 1-002 - Regularize headers, change from IO STS to EDT$$G IOFI NFND ,
47 0047 1 and from SY_EXIT to EDT$$FATAL IOERR . JBS 19-Feb-1981
48 0048 1 1-003 - Fix module name and file name. JBS 04-Mar-1981
49 0049 1 1-004 - Change entry point name from EDT to EDT$$MAIN_ENTRY . JBS 26-Mar-1981
50 0050 1 1-005 - Change E VERSION to EDT$$T VERSION. JBS 30-Mar-1981
51 0051 1 1-006 - Change EDT$$T VERSION to EDT$$, VER . JBS 30-Mar-1981
52 0052 1 1-007 - Correct a spelling error in EDT$$OPN_IFIDEF . JBS 02-Apr-1981
53 0053 1 1-008 - Use the new message codes. JBS 04-Aug-1981
54 0054 1 1-009 - Fix reading startup file. JBS 16-Aug-1981
55 0055 1 1-010 - Implement SET NOFNF. TMV 11-Sep-1981
56 0056 1 1-011 - Correct edit history. JBS 01-Oct-1981
57 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$$fiofn err. STS 26-Feb-1982
67 0067 1 1-022 - Add callable literals. STS 08-Mar-1982
68 0068 1 1-023 - Enable a handler. STS 09-Mar-1982
69 0069 1 1-024 - Remove the version number to a require file. JBS 09-Mar-1982
70 0070 1 1-025 - Check and set re-entry flag. STS 11-Mar-1982
71 0071 1 1-026 - Change EDT$$G_CMD and EDT$$G_JOU to EDT$$V_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
91 0091 1 mode, but do leave it for change mode to show. JBS 23-Jul-1982
92 0092 1 1-045 - Check the status returned by EDT$$EXE_CMD. JBS 28-Jul-1982
93 0093 1 1-046 - Add XLATE routine. STS 13-Aug-1982
94 0094 1 1-047 - Make use of the literals for the options word masks. STS 17-Aug-1982
95 0095 1 1-048 - Change the interface to EDT$$SET_COMFNAM. JBS 23-Aug-1982
96 0096 1 1-049 - Fix up references to EDT$M_bits. JBS 23-Aug-1982
97 0097 1 --
98 0098 1
```

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

EDTSMAIN
V04-000

EDTSMAIN - main entry point
Declarations

~~16-Sep-1984~~ ⁶ 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

:	157	0757	1	:
:	158	0758	1	:
:	159	0759	1	:
:	160	0760	1	:
:	161	0761	1	:

EXTERNAL REFERENCES:
Inside the routine.
-

```

163      0762 1 %SBTTL 'EDT$EDIT - Main entry point'
164      0763 1
165      0764 1 GLOBAL ROUTINE EDT$EDIT          ! Main entry point
166      0765 1
167      L 0766 1 %IF %BLISS (BLISS32)
168      0767 1 %THEN
169      0768 1     (INP_DESC,          ! Name of the input file
170      0769 1     OUT_DESC,          ! Name of the output file
171      0770 1     CMD_DESC,          ! Name of the command file
172      0771 1     JOURNAL_DESC,     ! Name of the journal file
173      0772 1     OPTIONS_WRD,      ! Options bits
174      0773 1     FILE_ROUT,        ! File I/O routine
175      0774 1     WORK_ROUT,        ! Work file I/O routine
176      0775 1     XLATE_ROUT,      ! XLATE command routine
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 ! +
191      0790 1 ! passed variables are set up in global data on the 11's in order to
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 !     JOURNAL_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 !     NONE
211      0810 1 !
212      0811 1 ! IMPLICIT OUTPUTS:
213      0812 1 !
214      0813 1 !     EDT$SA_INP_NAM
215      0814 1 !     EDT$SA_OUT_NAM
216      0815 1 !     EDT$SA_CMD_NAM
217      0816 1 !     EDT$SA_COM_NAM
218      0817 1 !     EDT$SA_JOU_NAM
219      0818 1 !     EDT$SG_INP_NAMLEN

```

```

220 0819 1 | EDTSSG_OUT_NAMLEN
221 0820 1 | EDTSSG_CMD_NAMLEN
222 0821 1 | EDTSSG_JOU_NAMLEN
223 0822 1 | EDTSSG_LASTMSG
224 0823 1 | EDTSSA_CMD_BUF
225 0824 1 | EDTSSA_CMD_END
226 0825 1 |
227 0826 1 | ROUTINE VALUE:
228 0827 1 |
229 0828 1 |     Normally 1 or EDT$_EDITORABO
230 0829 1 |
231 0830 1 | SIDE EFFECTS:
232 0831 1 |
233 0832 1 |     Performs an edit.
234 0833 1 |
235 0834 1 | --
236 0835 1 |
237 0836 1 | BEGIN
238 0837 1 |
239 L 0838 2 | XIF XBLISS (BLISS32)
240 0839 2 | XTHEN
241 0840 2 |
242 0841 2 | BUILTIN
243 0842 2 |     ACTUALCOUNT;
244 0843 2 |
245 0844 2 | MAP
246 0845 2 |     FILE_ROUT : REF VECTOR [2],
247 0846 2 |     WORK_ROUT : REF VECTOR [2],           ! Contains address of work i/o routine and environment
248 0847 2 |     XLATE_ROUT : REF VECTOR [2],
249 0848 2 |     CMD_DESC : REF BLOCK [8, BYTE],
250 0849 2 |     JOU_DESC : REF BLOCK [8, BYTE],
251 0850 2 |     OUT_DESC : REF BLOCK [8, BYTE],
252 0851 2 |     INP_DESC : REF BLOCK [8, BYTE],
253 0852 2 |     OPTIONS_WRD : REF BITVECTOR [32];
254 0853 2 |
255 0854 2 | XFI
256 0855 2 |
257 0856 2 | EXTERNAL ROUTINE
258 0857 2 |     EDTSSCALLFIO,           ! sets up handler and calls file handling routine
259 0858 2 |     EDTSSXE_CMD,           ! Execute commands
260 0859 2 |     EDTSSTERMIN_RCOVMOD : NOVALUE,       ! Handle end of journal file during /RECOVER
261 0860 2 |     EDTSSINIT,             ! Do the initial processing
262 0861 2 |     EDTSSTERMIN,           ! Do the final processing
263 0862 2 |     EDTSSFIOPN_ERR : NOVALUE,           ! Print a message about a file error, with file name
264 0863 2 |     EDTSSTY_CURLN,         ! Type out the current line
265 0864 2 |     EDTSSFMT_MSG : NOVALUE,           ! Format and print a message
266 0865 2 |     EDTSSSET_COMFNAM;       ! Open and close command files
267 0866 2 |
268 L 0867 2 | XIF XBLISS (BLISS32)
269 0868 2 | XTHEN
270 0869 2 |
271 0870 2 | EXTERNAL ROUTINE
272 0871 2 |     EDT$XLATE,
273 0872 2 |     EDT$FILEIO,
274 0873 2 |     EDT$WORKIO,
275 0874 2 |     EDT$SHANDLER,
276 0875 2 |     STRSCOPY_DX,           ! for copying all kinds of strings

```



```

: 277 0876 2 STR$FREE1_DX; ! for dynamic descriptor deallocation
: 278 0877 2
: 279 0878 2 %FI
: 280 0879 2
: 281 0880 2 EXTERNAL
: 282 0881 2 EDT$SG_REENTRY, ! flag signifying re-entry
: 283 0882 2 EDT$SA_COM_NAM, ! User-specified command file name
: 284 0883 2 EDT$SA_CMD_NAM, ! Command file name
: 285 0884 2 EDT$SG_CMD_NAMLEN, ! Command file name length
: 286 0885 2 EDT$ST_CMD_NAM_DEF1 : VECTOR [, BYTE], ! First default command file name (counted ASCII string)
: 287 0886 2 EDT$ST_CMD_NAM_DEF2 : VECTOR [, BYTE], ! First default command file default name (counted ASCII string)
: 288 0887 2 ! First default command file default name (counted ASCII string)
: 289 0888 2 EDT$ST_CMD_NAM_DEF3 : VECTOR [, BYTE], ! Second default command file name (counted ASCII string)
: 290 0889 2 EDT$ST_CMD_NAM_DEF4 : VECTOR [, BYTE], ! Second default command file default name (counted ASCII string)
: 291 0890 2 ! Second default command file default name (counted ASCII string)
: 292 0891 2 EDT$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$SG_CC_DONE; ! 1 = control C actually aborted something
: 310 0909 2
: 311 L 0910 2 %IF %BLISS (BLISS32)
: 312 0911 2 %THEN
: 313 0912 2
: 314 0913 2 EXTERNAL
: 315 0914 2 EDT$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$SG_WORK_ENV; ! environment word to pass work routine
: 321 0920 2
: 322 0921 2 %FI
: 323 0922 2
: 324 P 0923 2 MESSAGES ((COMFILNEX, REENTRY, COMFILNOP, COMFILCLO, NOJNLFIL, INPFILNEX, NONSTDFIL, EDITORABO, ABOBYCC,
: 325 0924 2 (CTRC_IGN));
: 326 0925 2
: 327 L 0926 2 %IF %BLISS (BLISS32)
: 328 0927 2 %THEN
: 329 0928 2
: 330 0929 2 ENABLE
: 331 0930 2 EDT$SHANDLER (); ! on VAX enable a handler for error processing
: 332 0931 2
: 333 0932 2 %FI

```

: R
:
:

334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390

```

0933 LOCAL
0934 STATUS,
0935 FILE_DESC : BLOCK [8, BYTE],
0936 RMB_DESC : BLOCK [8, BYTE],
0937
0938
L 0939 %IF %BLISS (BLISS32)
0940 %THEN
0941 L_INP_DESC : BLOCK [8, BYTE],
0942 L_OUT_DESC : BLOCK [8, BYTE],
0943 L_CMD_DESC : BLOCK [8, BYTE],
0944 L_JOU_DESC : BLOCK [8, BYTE],
0945
0946 %FI
0947
0948 PRIMARY_FILE_CONDITION,
0949 STRING_LENGTH;
0950
0951 !+
0952 Check the re-entry flag, so that the caller cannot
0953 re-enter EDT. A code of 67 indicates the everything
0954 o.k., a code of 68 means this process has already
0955 entered EDT. -Return an error message
0956 -
0957
0958 IF (.EDT$$G_REENTRY EQL 67) THEN EDT$$G_REENTRY = 68 ELSE RETURN (EDT$REENTRY);
0959
0960 EDT$$G_EXITD = 0;
0961
0962 !+
0963 On VAX/VMS we have fields defined for descriptors.
0964 Fill them in.
0965 -
L 0966 %IF %BLISS (BLISS32)
0967 %THEN
0968 FILE_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
0969 FILE_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
0970 FILE_DESC [DSC$A_POINTER] = 0;
0971 FILE_DESC [DSC$W_LENGTH] = 0;
0972 RMB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
0973 RMB_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
0974 RMB_DESC [DSC$A_POINTER] = 0;
0975 RMB_DESC [DSC$W_LENGTH] = 0;
0976
0977 IF (ACTUALCOUNT () GTR 5) AND (.FILE_ROUT NEQ 0)
0978 THEN
0979 BEGIN
0980 EDT$$A_FILE_ROUT = .FILE_ROUT [0]; ! get address for fileio routine passed
0981 EDT$$G_FILE_ENV = .FILE_ROUT [1]; ! and environment word
0982 END
0983 ELSE
0984 BEGIN
0985 EDT$$A_FILE_ROUT = EDT$FILEIO;
0986 EDT$$G_FILE_ENV = 0;
0987 END;
0988
0989 IF (ACTUALCOUNT () GTR 6) AND (.WORK_ROUT NEQ 0)
0990 THEN

```

EDT
V04

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

```
448 1047 2
449 1048 2
450 1049 2
451 1050 2
452 1051 2
453 1052 2
454 1053 2
455 1054 2
456 1055 2
457 1056 2
458 1057 2
459 1058 2
460 1059 2
461 1060 2
462 1061 2
463 1062 2
464 1063 2
465 1064 2
466 1065 2
467 1066 2
468 1067 2
469 1068 2
470 1069 2
471 1070 2
472 1071 2
473 1072 2
474 1073 2
475 1074 2
476 1075 2
477 1076 2
478 1077 2
479 1078 2
480 1079 2
481 1080 2
482 1081 2
483 1082 2
484 1083 2
485 L 1084 2
486 1085 2
487 1086 2
488 1087 2
489 1088 2
490 1089 2
491 1090 2
492 1091 2
493 1092 2
494 1093 2
495 1094 2
496 1095 2
497 1096 2
498 1097 2
499 1098 2
500 1099 2
501 1100 2
502 1101 2
503 1102 2
504 1103 2

IF (ACTUALCOUNT () GTR 3) AND (.JOU_DESC NEQ 0) THEN STR$COPY_DX (L_JOU_DESC, .JOU_DESC);
EDT$SA_JOU_NAM = .L_JOU_DESC [DSC$A_POINTER];
EDT$SG_JOU_NAMLEN = .L_JOU_DESC [DSC$W_LENGTH];
+
- Copy the options word into global storage so the rest of EDT can test it.
IF (ACTUALCOUNT () GTR 4) AND (.OPTIONS_WRD NEQ 0)
THEN
EDT$SV_OPTIONS = ..OPTIONS_WRD
ELSE
EDT$SV_OPTIONS = 0;
%FI
+
- Do the initial processing.
PRIMARY_FILE_CONDITION = EDT$INIT ();
+
- If initialization failed, abort right now.
IF (.PRIMARY_FILE_CONDITION EQL 3) THEN RETURN (EDT$EDITORABO);
+
- If /NOCREATE was specified then exit if file was not found
IF ((.PRIMARY_FILE_CONDITION EQL 0) AND ((.EDT$SV_OPTIONS AND EDT$M_NOCREATE) NEQ 0))
THEN
BEGIN
IF .EDT$SG_FNF_MSGFLG THEN EDT$FMT_MSG (EDT$_INPFILNEX);
L %IF %BLISS (BLISS32)
%THEN
STR$FREE1_DX (FILE_DESC);
STR$FREE1_DX (RHB_DESC);
STR$FREE1_DX (L_INP_DESC);
STR$FREE1_DX (L_OUT_DESC);
STR$FREE1_DX (L_CMD_DESC);
STR$FREE1_DX (L_JOU_DESC);
%FI
EDT$SG_REENTRY = 67;
RETURN (1);
END;
! set up reentry flag so we can call EDT again
+
- Open the journal file. If recovery was specified, open it so we can read
from it.
BEGIN
```

```

505 1104 LOCAL
506 1105     RHB_LEN;
507 1106
508 1107     RHB_LEN = %CHARCOUNT (JOU_DEFLT);
509 1108     STRING_DESC (RHB_DESC, RHB_LEN, CH$PTR (UPLIT (JOU_DEFLT)));
510 1109     END;
511 1110
512 1111 IF .EDT$$G_RCOV_MOD
513 1112 THEN
514 1113     BEGIN
515 1114     STRING_DESC (FILE_DESC, EDT$$G_JOU_NAMLEN, .EDT$$A_JOU_NAM);
516 1115     STATUS = EDT$$CALCFIO (EDT$K_OPEN_IN_OUT, EDT$K_JOURNAL_FILE, FILE_DESC, RHB_DESC);
517 1116
518 1117     IF ( NOT .STATUS)
519 1118     THEN
520 1119         BEGIN
521 1120         EDT$$FIOPN_ERR (EDT$ NOJNLFIL, FILE_DESC);
522 1121         RETURN (EDT$_EDITORABO);
523 1122         END;
524 1123
525 1124     END
526 1125 ELSE
527 1126     +
528 1127     We are not in recovery mode; just open the journal file for write.
529 1128     -
530 1129
531 1130     IF ((.EDT$$V_OPTIONS AND EDT$M_NOJOURNAL) EQL 0)           ! NOT /NOJOURNAL
532 1131     THEN
533 1132         BEGIN
534 1133         STRING_DESC (FILE_DESC, EDT$$G_JOU_NAMLEN, .EDT$$A_JOU_NAM);
535 1134         STATUS = EDT$$CALCFIO (EDT$K_OPEN_OUTPUT_NGSEQ, EDT$K_JOURNAL_FILE, FILE_DESC, RHB_DESC);
536 1135         END;
537 1136
538 1137     +
539 1138     Make sure the journal file was opened.
540 1139     -
541 1140
542 1141     IF (( NOT .STATUS) AND (.EDT$$V_OPTIONS AND EDT$M_NOJOURNAL) EQL 0)
543 1142     THEN
544 1143         BEGIN
545 1144         EDT$$FIOPN_ERR (EDT$ NOJNLFIL, FILE_DESC);
546 1145         RETURN (EDT$_EDITORABO);
547 1146         END;
548 1147
549 1148     +
550 1149     Execute the startup file, if one was provided.  The startup file is always executed
551 1150     in line mode.
552 1151     -
553 1152     EDT$$G_EDIT_MOD = LINE_MODE;
554 1153
555 1154     IF ((.EDT$$V_OPTIONS AND EDT$M_NOCOMMAND) EQL 0)
556 1155     THEN
557 1156         BEGIN
558 1157
559 1158         IF (.EDT$$G_CMD_NAMLEN EQL 0)
560 1159         THEN
561 1160             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$$SET COMFNAM (.EDT$$A_CMD_NAM, .EDT$$G_CMD_NAMLEN, CH$PTR (EDT$$ST_CMD_NAM_DEF4, 1),
585 1184 3 CH$RCHAR (EDT$$ST_CMD_NAM_DEF4));
586 1185 3
587 1186 3 IF .STATUS
588 1187 3 THEN
589 1188 4 BEGIN
590 1189 4
591 1190 5 IF (EDT$$EXE_CMD (INP_COMMAND) EQL 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$$G_CC_DONE THEN EDT$$FMT_MSG (EDT$_ABOBYCC) ELSE EDT$$FMT_MSG (EDT$_CTRC__IGN);
599 1198 5
600 1199 5 EDT$$A_CMD_BUF = EDT$$ST_CMD_BUF;
601 1200 5 EDT$$A_CMD_END = .EDT$$A_CMD_BUF;
602 1201 5 CH$WCHAR ('XC'!, .EDT$$A_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$$G_EDIT_DFLTMOD = LINE_MODE;
607 1206 4 END;
608 1207 4
609 1208 4 END
610 1209 3 ELSE
611 1210 3
612 1211 4 IF ((.EDT$$V_OPTIONS AND EDT$M_COMMAND) NEQ 0)
613 1212 3 THEN
614 1213 4 BEGIN
615 1214 4 STRING_DESC (FILE_DESC, EDT$$G_CMD_NAMLEN, .EDT$$A_CMD_NAM);
616 1215 4 EDT$$FIOFN_ERR (
617 1216 4
618 1217 4 IF .EDT$$G_IOFI_NFND THEN EDT$_COMFILNEX ELSE EDT$_COMFILNOP, FILE_DESC);
```

```

: 619 1218 4
: 620 1219 4 RETURN (EDT$_EDITORABO);
: 621 1220 4 END;
: 622 1221 3
: 623 1222 3 END;
: 624 1223 3
: 625 1224 2
: 626 1225 2 + Make sure the command file is closed, and the name is null.
: 627 1226 2 -
: 628 1227 2 STATUS = EDT$$SET_COMFNAM (0, 0, 0, 0); ! Close the command file
: 629 1228 2
: 630 1229 2 IF ( NOT .STATUS) THEN RETURN (EDT$_EDITORABO);
: 631 1230 2
: 632 1231 2 +
: 633 1232 2 Use the default mode setting.
: 634 1233 2 -
: 635 1234 2 EDT$$G_EDIT_MOD = .EDT$$G_EDIT_DFLTMOD;
: 636 1235 2 +
: 637 1236 2 Give an error message if the primary file was not found or does not have standard format,
: 638 1237 2 unless we are about to enter change mode, in which case defer the message.
: 639 1238 2 -
: 640 1239 2
: 641 1240 2 CASE .PRIMARY_FILE_CONDITION FROM 0 TO 2 OF
: 642 1241 2 SET
: 643 1242 2
: 644 1243 2 [0] :
: 645 1244 2
: 646 1245 2 IF (.EDT$$G_EDIT_MOD EQL LINE_MODE)
: 647 1246 2 THEN
: 648 1247 2 BEGIN
: 649 1248 2
: 650 1249 2 IF .EDT$$G_FNF_MSGFLG THEN EDT$$FMT MSG (EDT$_INPFILNEX);
: 651 1250 2
: 652 1251 2 END
: 653 1252 2 ELSE
: 654 1253 2
: 655 1254 2 IF .EDT$$G_FNF_MSGFLG THEN EDT$$G_LASTMSG = EDT$_INPFILNEX;
: 656 1255 2
: 657 1256 2 [1] :
: 658 1257 2 BEGIN
: 659 1258 2 0
: 660 1259 2 END;
: 661 1260 2
: 662 1261 2 [2] :
: 663 1262 2
: 664 1263 2 IF (.EDT$$G_EDIT_MOD EQL LINE_MODE)
: 665 1264 2 THEN
: 666 1265 2 EDT$$FMT_MSG (EDT$_NONSTDFIL)
: 667 1266 2 ELSE
: 668 1267 2 EDT$$G_LASTMSG = EDT$_NONSTDFIL;
: 669 1268 2
: 670 1269 2 TES;
: 671 1270 2
: 672 1271 2 +
: 673 1272 2 If we are in recovery mode, execute the remainder of the journal file.
: 674 1273 2 (Part of the journal file may have been used to respond to SUBSTITUTE/QUERY
: 675 1274 2 commands in the startup file.)

```

```

676 1275 2  !-
677 1276 2  ~~~~~
678 1277 2  IF (.EDTSSG_RCOV_MOD)
679 1278 2  THEN
680 1279 2  BEGIN
681 1280 2  ~~~~~
682 1281 4  IF (EDT$SEXE_CMD (INP_JOURNAL) EQL 2)
683 1282 3  THEN
684 1283 4  BEGIN
685 1284 4  !+
686 1285 4  ! The recovery process was interrupted by a control C. Bail out.
687 1286 4  !-
688 1287 4  ~~~~~
689 1288 4  IF .EDTSSG_CC_DONE THEN RETURN (EDT$_ABOBYCC) ELSE RETURN (EDT$_CTRC_IGN);
690 1289 4  ~~~~~
691 1290 2  END;
692 1291 2  ~~~~~
693 1292 2  EDT$TERMIN_RCOVMOD ();
694 1293 2  END;
695 1294 2  ~~~~~
696 1295 2  !+
697 1296 2  ! If we are in line mode, then print the current line.
698 1297 2  !-
699 1298 2  ~~~~~
700 1299 2  IF (.EDTSSG_EDIT_MOD EQL LINE_MODE)
701 1300 2  THEN
702 1301 2  BEGIN
703 1302 2  EDT$STY_CURLN ();
704 1303 2  END;
705 1304 2  ~~~~~
706 1305 2  !+
707 1306 2  ! And execute the main stream of commands.
708 1307 2  !-
709 1308 2  EDT$SEXE_CMD (INP_TERM);
710 1309 2  !+
711 1310 2  ! Do the finalization processing.
712 1311 2  !-
713 1312 2  EDT$TERMIN ();
714 1313 2  !+
715 1314 2  ! On the VAX we need to deallocate the dynamic descriptors
716 1315 2  !-
717 1316 2  ~~~~~
718 L 1317 2  %IF %BLISS (BLISS32)
719 1318 2  %THEN
720 1319 2  STR$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  %FI
727 1326 2  ~~~~~
728 1327 2  EDTSSG_REENTRY = 67;
729 1328 2  RETURN (1);
730 1329 2  END;

```

```

! set up reentry flag so we can call EDT again
! of routine EDTSEDT

```



```

.TITLE EDTSMMAIN EDTSMMAIN - main entry point
.IDENT \V04-000\
.PSECT _EDT$CODE,NOWRT, SHR, PIC,2
00 00 00 3B 55 4F 4A 2E 0000 P.AAA: .ASCII \.JOU;\<0><0><0> ;
.EXTRN EDTSM_NOCREATE, EDTSM_NOJOURNAL
.EXTRN EDTSM_COMMAND, EDTSM_ROCOMMAND
.EXTRN EDTSK_OPEN_IN_OUT
.EXTRN EDTSK_OPEN_INPUT
.EXTRN EDTSK_INPUT_FILE
.EXTRN EDTSK_JOURNAL_FILE
.EXTRN EDTSK_OPEN_OUTPUT_NOSEQ
.EXTRN EDTSK_COMMAND_FILE
.EXTRN EDTSK_CLOSE, EDTSSCALLFIO
.EXTRN EDTSS$EXE_CMD, EDTSS$TERMIN_RCOVMOD
.EXTRN EDTSS$INIT, EDTSS$TERMIN
.EXTRN EDTSS$FIOPN_ERR, EDTSS$TY_CURLN
.EXTRN EDTSS$FMT_MSG, EDTSS$SET_COMFNAM
.EXTRN EDT$XLATE, EDT$FILEIO
.EXTRN EDT$WORKIO, EDTSS$HANDLER
.EXTRN STRSCOPY_DX, STR$FREE1_DX
.EXTRN EDTSSG_REENTRY, EDTSSA_CMD_NAM
.EXTRN EDTSSA_CMD_NAM, EDTSSG_CMD_NAMLEN
.EXTRN EDTSS$CMD_NAM_DEF1
.EXTRN EDTSS$CMD_NAM_DEF2
.EXTRN EDTSS$CMD_NAM_DEF3
.EXTRN EDTSS$CMD_NAM_DEF4
.EXTRN EDTSSG_EXITD, EDTSSA_JOU_NAM
.EXTRN EDTSSG_JOU_NAMLEN
.EXTRN EDTSSG_RCOV_MOD
.EXTRN EDTSSG_EDIT_MOD
.EXTRN EDTSSG_EDIT_DFLTMOD
.EXTRN EDTSSA_INP_NAM, EDTSSG_INP_NAMLEN
.EXTRN EDTSSG_FNF_MSGFLG
.EXTRN EDTSSA_OUT_NAM, EDTSSG_OUT_NAMLEN
.EXTRN EDTSSG_IOFT_NFND
.EXTRN EDTSSV_OPTIONS, EDTSSG_LASTMSG
.EXTRN EDTSSA_CMD_BUF, EDTSS$CMD_R11
.EXTRN EDTSSA_CMD_END, EDTSSG_CC_DONE
.EXTRN EDTSSA_FILE_ROUT
.EXTRN EDTSSG_FILE_ENV
.EXTRN EDTSSA_XLATE_ROUT
.EXTRN EDTSSG_XLATE_ENV
.EXTRN EDTSSA_WORK_ROUT
.EXTRN EDTSSG_WORK_ENV
.EXTRN EDT$COMFILNEX, EDT$REENTRY
.EXTRN EDT$COMFILNOP, EDT$COMFILCLO
.EXTRN EDT$NOJNLFIL, EDT$INPFILNEX
.EXTRN EDT$NONSTDFIL, EDT$EDITORABO
.EXTRN EDT$ABOBYCC, EDT$CTRC__IGN
.EXTRN STRSCOPY_R
OFFC 00000
5B 00000000G 00 9E 00002
.ENTRY EDTSEEDIT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 0767
MOVAB EDTSSA_CMD_NAM, R11 ;

```

	5A	00000000G	00	9E	00009	MOVAB	EDTSSG_WORK_ENV, R10		
	59	00000000G	00	9E	00010	MOVAB	EDTSSG_CMD_RAM_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		
	6D	042E	CF	DE	0003D	MOVAL	45\$ (FP)	0836	
00000043	8F	00000000G	00	D1	00042	CMPL	EDTSSG_REENTRY, #67	0957	
			0A	12	0004D	BNEQ	1\$		
00000000G	00	44	8F	9A	0004F	MOVZBL	#68, EDTSSG_REENTRY		
			08	11	00057	BRB	2\$		
	50	00000000G	8F	D0	00059	1\$: MOVL	#EDTS_REENTRY, R0		
			04	00	00060	RET			
		00000000G	00	D4	00061	2\$: CLRL	EDTSSG_EXITD	0959	
2C	AE	020E0000	8F	D0	00067	MOVL	#34471936, FILE_DESC	0970	
		30	AE	D4	0006F	CLRL	FILE_DESC+4	0969	
24	AE	020E0000	8F	D0	00072	MOVL	#34471936, RHB_DESC	0974	
		28	AE	D4	0007A	CLRL	RHB_DESC+4	0973	
	05		6C	91	0007D	CMPB	(APT), #5	0976	
			1A	1B	00080	BLEQU	3\$		
		18	AC	D5	00082	TSTL	FILE_ROUT		
			15	13	00085	BEQL	3\$		
	50	18	AC	D0	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	
			11	11	0009A	BRB	4\$	0976	
00000000G	00	00000000G	00	9E	0009C	3\$: MOVAB	EDT\$FILEIO, EDTSSA_FILE_ROUT	0984	
		00000000G	00	D4	000A7	CLRL	EDTSSG_FILE_ENV	0985	
	06		6C	91	000AD	4\$: CMPB	(AP), #6	0988	
			17	1B	000B0	BLEQU	5\$		
		1C	AC	D5	000B2	TSTL	WORK_ROUT		
			12	13	000B5	BEQL	5\$		
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		
			0D	11	000C7	BRB	6\$	0988	
00000000G	00	C0000000G	00	9E	000C9	5\$: MOVAB	EDT\$WORKIO, EDTSSA_WORK_ROUT	0996	
			6A	D4	000D4	CLRL	EDTSSG_WORK_ENV	0997	
	07		6C	91	000D6	6\$: CMPB	(AP), #7	1000	
			1A	1B	000D9	BLEQU	7\$		
		20	AC	D5	000DB	TSTL	XLATE_ROUT		
			15	13	000DE	BEQL	7\$		
	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	
			0D	11	000F3	BRB	8\$	1000	
00000000G	00	00000000G	00	9E	000F5	7\$: MOVAB	EDT\$XLATE, EDTSSA_XLATE_ROUT	1008	
			6A	D4	00100	CLRL	EDTSSG_WORK_ENV	1009	
	1C	AE	020E0000	8F	D0	00102	8\$: 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		
	67		02	FB	00113	CALLS	#2, STRSCOPY_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	
			18	AE	D4	0012E	CLRL	L_OUT_DESC+4	1028
	01		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	
	67		02	FB	00141	CALLS	#2, STR\$COPY_DX		
00000000G	00	18	AE	D0	00144	MOVL	L_OUT_DESC+4, EDT\$\$A_OUT_NAM	1032	
00000000G	00	14	AE	3C	0014C	MOVZWL	L_OUT_DESC, EDT\$\$G_OUT_NAMLEN	1033	
	0C	AE	020E0000	8F	D0	00154	MOVL	#34471936, 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	
	67		02	FB	0016F	CALLS	#2, STR\$COPY_DX		
00000000G	68	10	AE	D0	00172	MOVL	L_CMD_DESC+4, EDT\$\$A_CMD_NAM	1041	
	00	0C	AE	3C	00176	MOVZWL	L_CMD_DESC, EDT\$\$G_CMD_NAMLEN	1042	
	04	AE	020E0000	8F	D0	0017E	MOVL	#34471936, 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	
	67		02	FB	00199	CALLS	#2, STR\$COPY_DX		
00000000G	00	08	AE	D0	0019C	MOVL	L_JOU_DESC+4, EDT\$\$A_JOU_NAM	1050	
00000000G	00	04	AE	3C	001A4	MOVZWL	L_JOU_DESC, EDT\$\$G_JOU_NAMLEN	1051	
			04	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	BC	D0	001B6	MOVL	@OPTIONS_WRD, EDT\$\$V_OPTIONS	1058
				02	11	001BA	BRB	13\$	
			64	D4	001BC	CLRL	EDT\$\$V_OPTIONS	1060	
00000000G	00	00	FB	001BE	CALLS	#0, EDT\$\$INIT		1067	
	53		50	D0	001C5	MOVL	R0, PRIMARY_FILE_CONDITION		
	03		53	D1	001C8	CMPB	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	EDT\$\$V_OPTIONS, #EDTSM_NOCREATE		
			17	13	001DB	BEQL	16\$		
	0D	00000000G	00	E9	001DD	BL3C	EDT\$\$G_FNF_MSGFLG, 15\$	1082	
		00000000G	8F	DD	001E4	PUSHL	#EDT\$ INPFILNEX		
00000000G	00		01	FB	001EA	CALLS	#1, EDT\$\$FMT_MSG		
			024B	31	001F1	BRW	44\$	1086	
			6E	05	D0	001F4	MOVL	#5, RHB_LEN	1107
				CF	9F	001F7	PUSHAB	P.AAA	1108
		FD	04	AE	9F	001FB	PUSHAB	RHB_LEN	
			2C	AE	9F	001FE	PUSHAB	RHB_DESC	

68		03	FB	00201	CALLS	#3, STR\$COPY_R			
33	00000000G	00	E9	00204	BLBC	EDT\$\$G_RCOV_MOD, 17\$	1111		
	00000000G	00	DD	0020B	PUSHL	EDT\$\$A_JOU_NAM	1114		
	00000000G	00	9F	00211	PUSHAB	EDT\$\$G_JOU_NAMLEN			
	34	AE	9F	00217	PUSHAB	FILE_DESC			
68		03	FB	0021A	CALLS	#3, STR\$COPY_R			
	24	AE	9F	0021D	PUSHAB	RMB_DESC	1115		
	30	AE	9F	00220	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, EDT\$\$CALCFIO			
	52	50	DO	00236	MOVL	R0, STATUS			
	51	52	E8	00239	BLBS	STATUS, 20\$	1117		
		43	11	0023C	BRB	19\$	1120		
00000000G	8F	64	D3	0023E	17\$: BITL	EDT\$\$V_OPTIONS, #EDTSM_NOJOURNAL	1130		
		2E	12	00245	BNEQ	18\$			
	00000000G	00	DD	00247	PUSHL	EDT\$\$A_JOU_NAM	1133		
	00000000G	00	9F	0024D	PUSHAB	EDT\$\$G_JOU_NAMLEN			
	34	AE	9F	00253	PUSHAB	FILE_DESC			
68		03	FB	00256	CALLS	#3, STR\$COPY_R			
	24	AE	9F	00259	PUSHAB	RMB_DESC	1134		
	30	AE	9F	0025C	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, EDT\$\$CALCFIO			
	52	50	DO	00272	MOVL	R0, STATUS			
	15	52	E8	00275	18\$: BLBS	STATUS, 20\$	1141		
00000000G	8F	64	D3	00278	BITL	EDT\$\$V_OPTIONS, #EDTSM_NOJOURNAL			
		0C	12	0027F	BNEQ	20\$			
	2C	AE	9F	00281	19\$: PUSHAB	FILE_DESC	1144		
	00000000G	8F	DD	00284	PUSHL	#EDT\$K_NOJNLFIL			
		00EA	31	0028A	BRW	29\$			
00000000G	66	01	DO	0028D	20\$: MOVL	#1, EDT\$\$G_EDIT_MOD	1152		
	8F	64	D3	00290	BITL	EDT\$\$V_OPTIONS, #EDTSM_NOCOMMAND	1154		
		68	12	00297	BNEQ	24\$			
	50	00000000G	00	DO	00299	MOVL	EDT\$\$G_CMD_NAMLEN, R0	1158	
			3C	12	002A0	BNEQ	21\$		
	7E	00000000G	00	9A	002A2	MOVZBL	EDT\$\$T_CMD_NAM_DEF2, -(SP)	1169	
		00000000G	00	9F	002A9	PUSHAB	EDT\$\$T_CMD_NAM_DEF2+1	1168	
	7E	00000000G	00	9A	002AF	MOVZBL	EDT\$\$T_CMD_NAM_DEF1, -(SP)	1169	
		00000000G	00	9F	002B6	PUSHAB	EDT\$\$T_CMD_NAM_DEF1+1	1166	
00000000G	00	04	FB	002BC	CALLS	#4, EDT\$\$SET_COMFNAM	1169		
	52	50	DO	002C3	MOVL	R0, STATUS			
	2C	52	E8	002C6	BLBS	STATUS, 23\$	1172		
	7E	69	9A	002C9	MOVZBL	EDT\$\$T_CMD_NAM_DEF4, -(SP)	1178		
		01	A9	9F	002CC	PUSHAB	EDT\$\$T_CMD_NAM_DEF4+1	1177	
	7E	00000000G	00	9A	002CF	MOVZBL	EDT\$\$T_CMD_NAM_DEF3, -(SP)	1178	
		00000000G	00	9F	002D6	PUSHAB	EDT\$\$T_CMD_NAM_DEF3+1	1175	
			0A	11	002DC	BRB	22\$	1178	
	7E		69	9A	002DE	21\$: MOVZBL	EDT\$\$T_CMD_NAM_DEF4, -(SP)	1184	
		01	A9	9F	002E1	PUSHAB	EDT\$\$T_CMD_NAM_DEF4+1	1183	
			50	DD	002E4	PUSHL	R0	1184	
			6B	DD	002E6	PUSHL	EDT\$\$A_CMD_NAM		
00000000G	00	04	FB	002E8	22\$: CALLS	#4, EDT\$\$SET_COMFNAM			
	52	50	DO	002EF	MOVL	R0, STATUS			
	53	52	E9	002F2	BLBC	STATUS, 27\$	1186		
		02	DD	002F5	23\$: PUSHL	#2	1190		

00000000G	30	01	FB	002F7	CALLS	#1, EDT\$\$EXE_CMD	
	02	50	D1	002FE	CMPL	RO, #2	
	08	7D	12	00301	BNEQ	30\$	1197
00000000G	00	E9	00303	24\$:	BLBC	EDT\$\$G_CC_DONE, 25\$	
00000000G	00	8F	DD	0030A	PUSHL	#EDT\$_ABOBYCC	
	00	06	11	00310	BRB	26\$	
00000000G	00	8F	DD	00312	25\$:	PUSHL	#EDT\$_CTRC_IGN
00000000G	00	01	FB	00318	26\$:	CALLS	#1, EDT\$\$FMT_MSG
00000000G	00	00	9E	0031F	MOVAB	EDT\$\$T_CMD_BUF, EDT\$\$A_CMD_BUF	1199
00000000G	00	00	D0	0032A	MOVL	EDT\$\$A_CMD_BUF, EDT\$\$A_CMD_END	1200
	50	00	D0	00335	MOVL	EDT\$\$A_CMD_END, RO	1201
	60	21	90	0033C	MOVB	#33, (RO)	
00000000G	00	01	D0	0033F	MOVL	#1, EDT\$\$G_EDIT_DFLTMOD	1205
	00	38	11	00346	BRB	30\$	1186
00000000G	8F	64	D3	00348	27\$:	BITL	EDT\$\$V_OPTIONS, #EDTSM_COMMAND
		2F	13	0034F	BEQL	30\$	1211
		6B	DD	00351	PUSHL	EDT\$\$A_CMD_NAM	1214
	00000000G	00	9F	00353	PUSHAB	EDT\$\$G_CMD_NAMLEN	
	34	AE	9F	00359	PUSHAB	FILE_DESC	
	68	03	FB	0035C	CALLS	#3, STR\$COPY_R	
	2C	AE	9F	0035F	PUSHAB	FILE_DESC	1215
	00000000G	00	E9	00362	BLBC	EDT\$\$G_IOFI_NFND, 28\$	1217
	00000000G	8F	DD	00369	PUSHL	#EDT\$_COMFILNEX	
		06	11	0036F	BRB	29\$	
00000000G	00	8F	DD	00371	28\$:	PUSHL	#EDT\$_COMFILNOP
		02	FB	00377	29\$:	CALLS	#2, EDT\$\$FIOPN_ERR
		11	11	0037E	BRB	31\$	1219
		7E	7C	00380	30\$:	CLRQ	-(SP)
		7E	7C	00382	CLRQ	-(SP)	1227
00000000G	00	04	FB	00384	CALLS	#4, EDT\$\$SET_COMFNAM	
	52	50	D0	0038B	MOVL	RO, STATUS	
	08	52	E8	0038E	BLBS	STATUS, 32\$	1229
	50	8F	D0	00391	31\$:	MOVL	#EDT\$_EDITORABO, RO
		04	00398	RET			
	66	00	D0	00399	32\$:	MOVL	EDT\$\$G_EDIT_DFLTMOD, EDT\$\$G_EDIT_MOD
02	00	53	CF	003A0	CASEL	PRIMARY_FILE_CONDITION, #0, #2	1234
002D	004C	0006	003A4	33\$:	.WORD	34\$-33\$,-	1240
						39\$-33\$,-	
						36\$-33\$	
	50	00	D0	003AA	34\$:	MOVL	EDT\$\$G_FNF_MSGFLG, RO
	01	66	D1	003B1	CMPL	EDT\$\$G_EDIT_MOD, #1	1245
		0B	12	003B4	BNEQ	35\$	
	37	50	E9	003B6	BLBC	RO, 39\$	1249
	00000000G	8F	DD	003B9	PUSHL	#EDT\$_INPFILNEX	
		1B	11	003BF	BRB	37\$	
	2C	50	E9	003C1	35\$:	BLBC	RO, 39\$
00000000G	00	8F	D0	003C4	MOVL	#EDT\$_INPFILNEX, EDT\$\$G_LASTMSG	1254
		1F	11	003CF	BRB	39\$	1245
	01	66	D1	003D1	36\$:	CMPL	EDT\$\$G_EDIT_MOD, #1
		0F	12	003D4	BNEQ	38\$	1263
	00000000G	8F	DD	003D6	PUSHL	#EDT\$_NONSTDFIL	1265
00000000G	00	01	FB	003DC	37\$:	CALLS	#1, EDT\$\$FMT_MSG
		0B	11	003E3	BRB	39\$	
00000000G	00	8F	D0	003E5	38\$:	MOVL	#EDT\$_NONSTDFIL, EDT\$\$G_LASTMSG
	2C	00	E9	003F0	39\$:	BLBC	EDT\$\$G_RCOV_MOD, 42\$
		03	DD	003F7	PUSHL	#3	1277
00000000G	00	01	FB	003F9	CALLS	#1, EDT\$\$EXE_CMD	1281

: R

02		50	D1	00400	C MPL	R0, #2		
		17	12	00403	B NEQ	41\$		
08	00000000G	00	E9	00405	BLBC	EDT\$\$G_CC_DONE, 40\$	1288	
50	00000000G	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\$\$TERMIN_RCOVMOD	1292	
	01	66	D1	00423	42\$: C MPL	EDT\$\$G_EDIT_MOD, #1	1299	
		07	12	00426	B NEQ	43\$		
00000000G	00	00	FB	00428	CALLS	#0, EDT\$\$TY_CURLN	1302	
		7E	D4	0042F	43\$: CLRL	-(SP)	1308	
00000000G	00	01	FB	00431	CALLS	#1, EDT\$\$EXE_CMD		
00000000G	00	00	FB	00438	CALLS	#0, EDT\$\$TERMIN	1312	
		2C	AE	9F	0043F	44\$: PUSHAB	FILE_DESC	1319
	65	01	FB	00442	CALLS	#1, STR\$FREE1_DX		
		24	AE	9F	00445	PUSHAB	RMB_DESC	1320
	65	01	FB	00448	CALLS	#1, STR\$FREE1_DX		
		1C	AE	9F	0044B	PUSHAB	L_INP_DESC	1321
	65	01	FB	0044E	CALLS	#T, STR\$FREE1_DX		
		14	AE	9F	00451	PUSHAB	L_OUT_DESC	1322
	65	01	FB	00454	CALLS	#T, STR\$FREE1_DX		
		0C	AE	9F	00457	PUSHAB	L_CMD_DESC	1323
	65	01	FB	0045A	CALLS	#T, STR\$FREE1_DX		
		04	AE	9F	0045D	PUSHAB	L_JOU_DESC	1324
	65	01	FB	00460	CALLS	#T, STR\$FREE1_DX		
00000000G	00	43	8F	9A	00463	MOVZBL	#67, EDT\$\$G_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	7E	04	AC	7D	00475	MOVQ	4(AP), -(SP)	
	00		03	FB	00479	CALLS	#3, EDT\$\$HANDLER	
			04	00480	RET			

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

: 731 1330 1
: 732 1331 1 !<BLF/PAGE>

EDTSMIN
V04-000

EDTSMIN - main entry point
EDTSEDIT - Main entry point

1 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;1

Page 21
(4)

**F

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

: of module EDTSMIN

PSECT SUMMARY

Name	Bytes	Attributes
_EDTSCODE	1161	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	6	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	00:04.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:MAIN/OBJ=OBS\$:MAIN MSRCS\$:MAIN.BLI/UPDATE=(ENHS\$: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

