


```

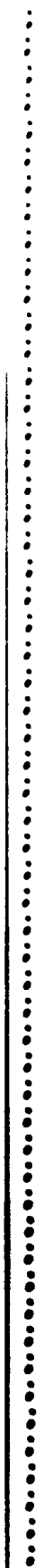
CCCCCCCC HH    HH  MM    MM  EEEEEEEEEE XX    XX  VV    VV  EEEEEEEEEE RRRRRRRR 88888888
CCCCCCCC HH    HH  MM    MM  EEEEEEEEEE XX    XX  VV    VV  EEEEEEEEEE RRRRRRRR 88888888
CC        HH    HH  MMMM  MMMM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CC        HH    HH  MMMM  MMMM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CC        HH    HH  MM    MM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CC        HH    HH  MM    MM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CC        HHHHHHHHHH MM    MM  EEEEEEEEEE XX    XX  VV    VV  EEEEEEEEEE RRRRRRRR 88888888
CC        HHHHHHHHHH MM    MM  EEEEEEEEEE XX    XX  VV    VV  EEEEEEEEEE RRRRRRRR 88888888
CC        HH    HH  MM    MM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CC        HH    HH  MM    MM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CC        HH    HH  MM    MM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CC        HH    HH  MM    MM  EE          XX    XX  VV    VV  EE          RR    RR  88    88
CCCCCCCC HH    HH  MM    MM  EEEEEEEEEE XX    XX  VV    VV  EEEEEEEEEE RR    RR  88888888
CCCCCCCC HH    HH  MM    MM  EEEEEEEEEE XX    XX  VV    VV  EEEEEEEEEE RR    RR  88888888

```

```

LL        IIIIII  SSSSSSSS
LL        IIIIII  SSSSSSSS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SSSSSS
LL        II      SSSSSS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SS
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```



```
1 0001 0 %TITLE 'EDT$CHMEXVERB - execute certain change-mode commands'  
2 0002 0 MODULE EDT$CHMEXVERB ( ! Execute certain change-mode commands  
3 0003 0 IDENT = 'V04-000' ! File: CHMEXVERB.BLI Edit: JBS1034  
4 0004 0 ) =  
5 0005 1 BEGIN  
6 0006 1  
7 0007 1 *****  
8 0008 1 *  
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
11 0011 1 * ALL RIGHTS RESERVED. *  
12 0012 1 *  
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
18 0018 1 * TRANSFERRED. *  
19 0019 1 *  
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
22 0022 1 * CORPORATION. *  
23 0023 1 *  
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
26 0026 1 *  
27 0027 1 *  
28 0028 1 *****  
29 0029 1  
30 0030 1  
31 0031 1 **  
32 0032 1 FACILITY: EDT -- The DEC Standard Editor  
33 0033 1  
34 0034 1 ABSTRACT:  
35 0035 1  
36 0036 1 This module executes those change mode commands which  
37 0037 1 take an entity.  
38 0038 1  
39 0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant  
40 0040 1  
41 0041 1 AUTHOR: Bob Kushlis, CREATION DATE: Unknown  
42 0042 1  
43 0043 1 MODIFIED BY:  
44 0044 1  
45 0045 1 1-001 - Original. DJS 04-Feb-1981. This module was created by  
46 0046 1 extracting routine EXECUTE_VERB from module CHANGE.BLI.  
47 0047 1 1-002 - Regularize headers. JBS 02-Mar-1981  
48 0048 1 1-003 - Fix bug in processing the SR entity. As preparation for  
49 0049 1 doing a string search, the cursor was moved to the left, the  
50 0050 1 search was done, and the cursor was moved back to the right  
51 0051 1 if the search failed. Since the value returned from EDT$SCS_LEFT  
52 0052 1 was ignored, it was possible to end up at a different place  
53 0053 1 than you began. DJS 04-Mar-1981  
54 0054 1 1-004 - Correct minor errors in the headers. JBS 12-Mar-1981  
55 0055 1 1-005 - Build in T.ADV AND T.BACK. JBS 31-Mar-1981  
56 0056 1 1-006 - Use the ASSERT macro. JBS 01-Jun-1981  
57 0057 1 1-007 - Repair a comment. JBS 02-Jun-1981
```

```
58 0058 1 1-008 - Correct the test for legit entities. JBS 02-Jun-1981
59 0059 1 1-009 - Use new message codes. JBS 04-Aug-1981
60 0060 1 1-010 - Add a check for change case verbs. STS 21-Sep-1981
61 0061 1 1-011 - Add search and select verb. STS 24-Sep-1981
62 0062 1 1-012 - Add return value for end of journal file. JBS 02-Oct-1981
63 0063 1 1-013 - Add new word type and para types. STS 26-Oct-1981
64 0064 1 1-014 - Remove an unused external reference. JBS 18-Dec-1981
65 0065 1 1-015 - Fix problem with repeated CHGCSR's and run through PRETTY. JBS 29-Dec-1981
66 0066 1 1-016 - Fix bug in ADV comparison - string length incorrect. SMB 30-Jan-1982
67 0067 1 1-017 - Worry about string truncation. JBS 05-May-1982
68 0068 1 1-018 - Set a flag if control C actually aborts something. JBS 24-May-1982
69 0069 1 1-019 - Don't try to keep moving left if at beginning of buffer. STS 21-Jun-1982
70 0070 1 1-020 - Change string matching. JBS 16-Jul-1982
71 0071 1 1-021 - Remove some redundant code. JBS 23-Jul-1982
72 0072 1 1-022 - New screen update logic. JBS 13-Sep-1982
73 0073 1 1-023 - Make call to edt$$stst_eob in line. STS 22-Sep-1982
74 0074 1 1-024 - Remove EDT$$G_LN_NO for new screen update logic. JBS 29-Sep-1982
75 0075 1 1-025 - Correct a typo in edit 1-024. JBS 30-Sep-1982
76 0076 1 1-026 - Make tests for ADV and BACK be case insensitive. JBS 10-Nov-1982
77 0077 1 1-027 - Change call to EDT$$MRK_LNCHG. JBS 27-Dec-1982
78 0078 1 1-028 - Plant traps for a problem involving unreasonable character position. JBS 27-Dec-1982
79 0079 1 1-029 - Improve those traps. JBS 28-Dec-1982
80 0080 1 1-030 - Add conditional for WPS support. JBS 10-Feb-1983
81 0081 1 1-031 - Remove an ASSERT which causes an internal error. SMB 11-Feb-1983
82 0082 1 1-032 - Don't look for ADV or BACK beyond the end of the command. JBS 04-Mar-1983
83 0083 1 1-033 - Fix -NL, and decrease size on PDP-11s by removing consistency checking. JBS 03-Jun-1983
84 0084 1 1-034 - Fix DESEL followed by SSEL in the same keystroke. JBS 20-Sep-1983
85 0085 1 --
86 0086 1
```

```

: 88      0087 1 %SBTTL 'Declarations'
: 89      0088 1
: 90      0089 1 | TABLE OF CONTENTS:
: 91      0090 1 |
: 92      0091 1 |
: 93      0092 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 94      0531 1
: 95      0532 1 FORWARD ROUTINE
: 96      0533 1     EDT$SEXE_CHMCM2,
: 97      0534 1     CHANGE_CASE : NOVALUE;
: 98      0535 1
: 99      0536 1 |
100     0537 1 | INCLUDE FILES:
101     0538 1 |
102     0539 1 |
103     0540 1 REQUIRE 'EDTSRC:EDTREQ';
104     0675 1
105     0676 1 LIBRARY 'EDTSRC:SUPPORTS';
106     0677 1
107     0678 1 |
108     0679 1 | MACROS:
109     0680 1 |
110     0681 1 |     NONE
111     0682 1 |
112     0683 1 | EQUATED SYMBOLS:
113     0684 1 |
114     0685 1 |
115     L 0686 1 %IF %BLISS (BLISS16)
116     U 0687 1 %THEN
117     U 0688 1
118     U 0689 1 LITERAL
119     U 0690 1     EXTRA_ASSERTS = 0;
120     U 0691 1
121     0692 1 %ELSE
122     0693 1
123     0694 1 LITERAL
124     0695 1     EXTRA_ASSERTS = 1;
125     0696 1
126     0697 1 %FI
127     0698 1 |
128     0699 1 |
129     0700 1 | OWN STORAGE:
130     0701 1 |
131     0702 1 |     NONE
132     0703 1 |
133     0704 1 | EXTERNAL REFERENCES:
134     0705 1 |
135     0706 1 |     In the routine

```

```

! Execute the verbs which take an entity specification
! Change case in a range

```

```

137 0707 1 %SBTTL 'EDT$$EXE_CHMCMD2 - execute certain change-mode commands'
138 0708 1
139 0709 1 GLOBAL ROUTINE EDT$$EXE_CHMCMD2 (           ! Execute certain change-mode commands
140 0710 1     ENTITY,                               ! the entity to use
141 0711 1     COUNT,                               ! the number of entities to include
142 0712 1     VERB                               ! the command to execute
143 0713 1     ) =
144 0714 1
145 0715 1 ++
146 0716 1 FUNCTIONAL DESCRIPTION:
147 0717 1
148 0718 1     This routine executes verbs which take an entity specification.
149 0719 1
150 0720 1     First it isolates the text defined by the entity type and the entity
151 0721 1     count which are passed as parameters; then performs the verb on the
152 0722 1     selected text. It then executes the verb on the text it has selected.
153 0723 1
154 0724 1 FORMAL PARAMETERS:
155 0725 1
156 0726 1     ENTITY           the entity to use
157 0727 1
158 0728 1     COUNT           the number of entities to include
159 0729 1
160 0730 1     VERB           the command to execute.
161 0731 1
162 0732 1 IMPLICIT INPUTS:
163 0733 1
164 0734 1     EDT$$T_DEL_CH
165 0735 1     EDT$$T_DEL_LN
166 0736 1     EDT$$T_DEL_WD
167 0737 1     EDT$$A_CMD_BUF
168 0738 1     EDT$$A_CMD_END
169 0739 1     EDT$$A_SEA_STR
170 0740 1     EDT$$A_SEL_BUF
171 0741 1     EDT$$A_SEL_POS
172 0742 1     EDT$$G_SEA_LEN
173 0743 1     EDT$$G_VERT
174 0744 1     EDT$$A_US_ENT
175 0745 1     EDT$$G_SEA_BEG
176 0746 1     EDT$$A_CUR_BUF
177 0747 1     EDT$$G_TI_TYP
178 0748 1     EDT$$T_LN_BUF
179 0749 1     EDT$$A_LN_END
180 0750 1     EDT$$G_LN_LEN
181 0751 1     EDT$$A_LN_PTR
182 0752 1     EDT$$G_SEA_STRLIN
183 0753 1     EDT$$T_SEA_STR
184 0754 1     EDT$$G_PARTYP
185 0755 1     EDT$$A_WK_LN
186 0756 1     EDT$$A_OLD_SEL
187 0757 1
188 0758 1 IMPLICIT OUTPUTS:
189 0759 1
190 0760 1     EDT$$G_DIR
191 0761 1     EDT$$G_VERT
192 0762 1     EDT$$A_LN_PTR
193 0763 1     EDT$$A_CMD_BUF

```

```

194 0764 1 | EDT$$G_DEL_CHLEN
195 0765 1 | EDT$$G_DEL_WDLEN
196 0766 1 | EDT$$G_DEL_LNLEN
197 0767 1 | EDT$$G_CAS_TYP
198 0768 1 | EDT$$A_ALT_BUF
199 0769 1 | EDT$$A_CUR_BUF
200 0770 1 | EDT$$G_LN_CEN
201 0771 1 | EDT$$G_CC_DONE
202 0772 1 | EDT$$G_SCR_REBUILD
203 0773 1 |
204 0774 1 | ROUTINE VALUE:
205 0775 1 |
206 0776 1 |     1 = ok, 0 = hit a boundry, 2 = end of journal file
207 0777 1 |
208 0778 1 | SIDE EFFECTS:
209 0779 1 |
210 0780 1 |     MANY
211 0781 1 |
212 0782 1 | --
213 0783 1 |
214 0784 2 | BEGIN
215 0785 2 |
216 0786 2 | EXTERNAL ROUTINE
217 0787 2 | EDT$$FND_BSEN : NOVALUE, | Find the beginning of a sentence
218 0788 2 | EDT$$FND_BWD, | Move backwards to the beginning of a word
219 0789 2 | EDT$$DEL_TXTLN : NOVALUE, | Delete whole lines of text
220 0790 2 | EDT$$MSG_BELL : NOVALUE, | Output a message to the terminal with a warning bell
221 0791 2 | EDT$$CHK_CC, | Check to see if a CTRL/C has been typed
222 0792 2 | EDT$$TADJ_CMD : NOVALUE, | Execute the TADJ command
223 0793 2 | EDT$$FND_EWD, | Find the end of a word
224 0794 2 | EDT$$FND_ENT, | Search for a page or paragraph entity
225 0795 2 | EDT$$STR_SEACMD, | Search for a specific string
226 0796 2 | EDT$$FILC_TXT, | Do a text fill
227 0797 2 | EDT$$INS_MOD, | Process no-keypad insertion
228 0798 2 | EDT$$CS_BOTBUF, | Move to bottom of buffer
229 0799 2 | EDT$$RPL_CHGDLN, | Declare current line as changed
230 0800 2 | EDT$$CS_DWN, | Move down a line
231 0801 2 | EDT$$GET_TXTLN, | Get current line in line buffer
232 0802 2 | EDT$$CS_LEFT, | Move left a character
233 0803 2 | EDT$$CS_RIGHT, | Move right a character
234 0804 2 | EDT$$CS_TOP, | Move to top of buffer
235 0805 2 | EDT$$CS_UP, | Move up a line
236 0806 2 | EDT$$TST_ONSTR, | Compare the current character position with a string descriptor
237 0807 2 | EDT$$RPOS : NOVALUE, | Restore the saved buffer position
238 0808 2 | EDT$$SAV_BUFPOS : NOVALUE, | Save the current buffer position
239 0809 2 | EDT$$SAV_DELTXT : NOVALUE, | Save a deleted word or line in an undelete buffer
240 0810 2 | EDT$$SC_MATCHCOL, | Match column
241 0811 2 | EDT$$SC_CPUNEWCOLPOS, | Compute new column
242 0812 2 | EDT$$SEC_RNGPOS, | Compare the select line with the current line
243 0813 2 | EDT$$FND_SEDLIM, | Look for a sentence delimiter
244 0814 2 | EDT$$STR_CMP, | Compare two strings of equal length
245 0815 2 | EDT$$SET_SEASTR, | Setup string search buffer
246 0816 2 | EDT$$WF_BOT, | Go to bottom of file
247 0817 2 | EDT$$WF_CLRBUF, | Clear a buffer
248 0818 2 | EDT$$END_INS, | End an insert sequence
249 0819 2 | EDT$$DEL_CURLN, | Delete a line from buffer
250 0820 2 | EDT$$RPL_LN, | Replace a line in text buffer

```

```

251      0821      2      EDT$$RD_PRVLN,      ! Move backward a line
252      0822      2      EDT$$RD_CURLN,      ! Get the current line
253      0823      2
254      L 0824      2      %IF SUPPORT_WPS
255      0825      2      %THEN
256      0826      2      EDT$$FND_WPARA,      ! find a wps paragraph
257      0827      2      %FI
258      0828      2
259      0829      2      EDT$$START_INS;      ! Start an insert sequence
260      0830      2
261      0831      2      EXTERNAL
262      0832      2
263      L 0833      2      %IF SUPPORT_WPS
264      0834      2      %THEN
265      0835      2      EDT$$G_CAS_TYP,
266      0836      2      %FI
267      0837      2
268      0838      2      EDT$$T_DEL_CH : BLOCK      ! Deleted character buffer.
269      0839      2      [CH$ALLOCATION (2)],
270      0840      2      EDT$$G_DEL_CHLEN,      ! Length of deleted character buffer
271      0841      2      EDT$$T_DEL_LN : BLOCK      ! Deleted line buffer.
272      0842      2      [CH$ALLOCATION (257)],
273      0843      2      EDT$$G_DEL_LNLEN,      ! Deleted line length.
274      0844      2      EDT$$G_DIR,      ! The current direction.
275      0845      2      EDT$$T_DEL_WD : BLOCK      ! Deleted word buffer.
276      0846      2      [CH$ALLOCATION (81)],
277      0847      2      EDT$$G_DEL_WDLEN,      ! Length of del word string.
278      0848      2      EDT$$A_CMD_BUF,      ! Command string pointer
279      0849      2      EDT$$A_CMD_END,      ! Command string end
280      0850      2      EDT$$A_SEA_STR,      ! Address of search string.
281      0851      2      EDT$$A_SEL_BUF,      ! Pointer to select buffer.
282      0852      2      EDT$$A_OLD_SEL,      ! Select buffer before last screen update
283      0853      2      EDT$$G_SCR_REBUILD,      ! 1 = repaint the screen
284      0854      2      EDT$$L_SEL_LN : LN_BLOCK,      ! Relative line number of select.
285      0855      2      EDT$$A_SEL_POS,      ! select position.
286      0856      2      EDT$$G_SEA_LEN,      ! Length of search string.
287      0857      2      EDT$$G_VERT,      ! Last entity was VERT flag.
288      0858      2      EDT$$A_ALT_BUF : REF TBCB_BLOCK,      ! Alternate buffer used for cut/paste.
289      0859      2      EDT$$A_US_ENT : VECTOR,      ! Pointers to user defined entities
290      0860      2      EDT$$G_SEA_BEG,      ! Leave search at begining if on
291      0861      2      EDT$$A_CUR_BUF : REF TBCB_BLOCK,      ! The current buffer tbc
292      0862      2      EDT$$G_TI_TYP,      ! Terminal type.
293      0863      2      EDT$$T_LN_BUF,      ! Current line buffer
294      0864      2      EDT$$A_LN_END,      ! End of current line pointer
295      0865      2      EDT$$G_LN_LEN,      ! Length of current line
296      0866      2      EDT$$A_LN_PTR,      ! Current character pointer
297      0867      2      EDT$$G_SEA_STRLEN,      ! Length of current search string
298      0868      2      EDT$$T_SEA_STR,      ! Current search string
299      0869      2
300      L 0870      2      %IF SUPPORT_WPS
301      0871      2      %THEN
302      0872      2      EDT$$G_PARTYP,      ! type of paragraph
303      0873      2      %FI
304      0874      2
305      0875      2      EDT$$A_WK_LN : REF LIN_BLOCK,      ! Current line pointer
306      0876      2      EDT$$Z_EOB_LN,
307      0877      2      EDT$$G_CC_DCNE,      ! Set to 1 if control C actually aborts something

```



```

308 0878 2      EDT$$L_LN00 : LN_BLOCK;          ! Line number with value 1
309 0879 2
310 0880 2      MESSAGES ((NOSELNAN, SELALRACT, INVENT, ATTCUTAPP, STRNOTFND, TOPOFBUF, BOTOFBUF, INVSTR, BADRANGE));
311 0881 2
312 0882 2      LOCAL
313 0883 2          START_POS : POS_BLOCK,          ! Position of start of text.
314 0884 2          END_POS : POS_BLOCK,          ! Position of end of text.
315 0885 2          ORIG_LNO : LN_BLOCK,          ! Original record number
316 0886 2          NUM_LINES : LN_BLOCK,          ! Number of lines in the range.
317 0887 2          SR,                          ! Set if range was a select range.
318 0888 2          NC,                          ! Number of characters included in this entity.
319 0889 2          SUCCEED;                      ! Set to zero if we hit a boundary.
320 0890 2
321 0891 2      SUCCEED = 1;
322 0892 2      SR = 0;
323 0893 2      NC = 0;
324 0894 2      !+
325 0895 2      !- Setup the search string if it is a quoted string.
326 0896 2
327 0897 2
328 0898 2      IF (.ENTITY EQL ENT_K_QUOTE)
329 0899 2      THEN
330 0900 2          BEGIN
331 0901 2
332 0902 2          IF ( NOT EDT$$SET_SEASTR (.EDT$$A_SEA_STR, .EDT$$G_SEA_LEN))
333 0903 2          THEN
334 0904 2              BEGIN
335 0905 2                  SUCCEED = 0;
336 0906 2                  EDT$$MSG_BELL (EDT$_INVSTR);
337 0907 2                  RETURN (.SUCCEED);
338 0908 2              END;
339 0909 2
340 0910 2      !+
341 0911 2      !- If the next command is advance or backup, set the direction now
342 0912 2      so we can use the advance and backup keys to terminate the search
343 0913 2      string like KED does.
344 0914 2
345 0915 2
346 0916 2      WHILE ((CH$RCHAR (.EDT$$A_CMD_BUF) EQL %C' ') AND ((.EDT$$A_CMD_BUF + 3) LEQA .EDT$$A_CMD_END)) DO
347 0917 2          EDT$$A_CMD_BUF = CH$PCUS (.EDT$$A_CMD_BUF, 1);
348 0918 2
349 0919 2      IF ((.EDT$$A_CMD_BUF + 3) LEQA .EDT$$A_CMD_END)
350 0920 2      THEN
351 0921 2          BEGIN
352 0922 2
353 0923 2          IF EDT$$STR_CMP (.EDT$$A_CMD_BUF, UPLIT (BYTE ('BACK')), 4, 3)
354 0924 2          THEN
355 0925 2              EDT$$G_DIR = DIR_BACKWARD
356 0926 2          ELSE
357 0927 2
358 0928 2              IF EDT$$STR_CMP (.EDT$$A_CMD_BUF, UPLIT (BYTE ('ADV')), 3, 3) THEN EDT$$G_DIR = DIR_FORWARD;
359 0929 2
360 0930 2          END;
361 0931 2
362 0932 2      END;
363 0933 2
364 0934 2      !+

```

```
365 0935 2 ! To make the line, word, sent, etc entities work properly for deletes
366 0936 2 ! we must first move to the beginning of the entity. This does not
367 0937 2 ! apply for moves.
368 0938 2 !
369 0939 2 !
370 0940 2 !
371 0941 2 ! IF ((.VERB NEQ VERB_K_MOVE) AND
372 0942 2 ! (.VERB NEQ VERB_K_CHGC) AND
373 0943 2 ! (.VERB NEQ VERB_K_CHGU) AND
374 0944 2 ! (.VERB NEQ VERB_K_CHGL))
375 0945 2 ! THEN
376 0946 2 ! CASE .ENTITY/2 FROM ENT_K_CHAR/2 TO LAST_K_ENT/2 OF
377 0947 2 ! SET
378 0948 2 !
379 0949 2 ! [ENT K LINE/2] :
380 0950 2 ! EDT$$A_LN_PTR = CHSPTR (EDT$$T_LN_BUF);
381 0951 2 !
382 0952 2 ! [ENT K WORD/2] :
383 0953 2 ! EDT$$FND_BWD (0);
384 0954 2 !
385 0955 2 ! [ENT K SEN/2] :
386 0956 2 ! EDT$$FND_BSEN ();
387 0957 2 !
388 0958 2 ! [ENT K PAGE/2] :
389 0959 2 ! BEGIN
390 0960 2 ! EDT$$CS_RIGHT ();
391 0961 2 ! SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [2 + (.ENTITY EQL ENT_K_PAGE)], DIR_BACKWARD, 1);
392 0962 2 !
393 0963 2 ! IF (.SUCCEED EQL 2) THEN SUCCEED = 0 ELSE SUCCEED = 1;
394 0964 2 !
395 0965 2 ! END;
396 0966 2 !
397 0967 2 ! [ENT K PAR/2] :
398 0968 2 ! BEGIN
399 0969 2 ! EDT$$CS_RIGHT ();
400 0970 2 !
401 0971 2 ! %IF SUPPORT_WPS
402 0972 2 ! %THEN
403 0973 2 !
404 0974 2 ! IF (.EDT$$G_PARTYP EQL WSPARA)
405 0975 2 ! THEN
406 0976 2 ! SUCCEED = EDT$$FND_WPARA (DIR_BACKWARD)
407 0977 2 ! ELSE
408 0978 2 ! %FI
409 0979 2 !
410 0980 2 ! BEGIN
411 0981 2 ! SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [2], DIR_BACKWARD, 1);
412 0982 2 !
413 0983 2 ! IF (.SUCCEED EQL 2) THEN SUCCEED = 0 ELSE SUCCEED = 1;
414 0984 2 !
415 0985 2 ! END;
416 0986 2 !
417 0987 2 ! END;
418 0988 2 !
419 0989 2 ! [INRANGE] :
420 0990 2 ! BEGIN
421 0991 2 ! 0
```

```
422      0992      2      END;
423      0993      2
424      L 0994      2      %IF EXTRA_ASSERTS
425      0995      2      %THEN
426      0996      2
427      0997      2      [OUTRANGE] :
428      0998      2      ASSERT (0);
429      0999      2      %FI
430      1000      2
431      1001      2      TES;
432      1002      2
433      1003      2      !+
434      1004      2      ! Save the original position in the buffer.
435      1005      2      !-
436      1006      2      EDT$$SAV_BUFPOS (START_POS);
437      1007      2
438      L 1008      2      %IF EXTRA_ASSERTS
439      1009      2      %THEN
440      1010      2      ASSERT (.START_POS [POS_CHAR_POS] LEQU 255);
441      1011      2      %FI
442      1012      2
443      1013      2      MOVELINE (EDT$$A_CUR_BUF [TBCB_CUR_LIN], ORIG_LNO);
444      1014      2      !+
445      1015      2      ! Loop over the entity count moving by one entity each time.
446      1016      2      !-
447      1017      2
448      1018      2      INCR I FROM 1 TO .COUNT DO
449      1019      2      BEGIN
450      1020      3
451      1021      4      IF (EDT$$CHK_CC () NEQ 0)
452      1022      3      THEN
453      1023      4      BEGIN
454      1024      4      EDT$$G_CC_DONE = 1;
455      1025      4      EXITLOOP;
456      1026      4      END;
457      1027      3
458      1028      2      !+
459      1029      2      ! Process one entity of the specified type.
460      1030      2      !-
461      1031      3
462      1032      3      CASE .ENTITY + .EDT$$G_DIR FROM 1 TO LAST_K_ENT + 1 OF
463      1033      3      SET
464      1034      3
465      1035      3      [ENT_K_CHAR + DIR_FORWARD] :
466      1036      4      BEGIN
467      1037      4
468      1038      4      IF (SUCCEED = EDT$$CS_RIGHT ()) THEN NC = 1;
469      1039      4
470      1040      3      END;
471      1041      3
472      1042      3      [ENT_K_CHAR + DIR_BACKWARD] :
473      1043      4      BEGIN
474      1044      4
475      1045      4      IF (SUCCEED = EDT$$CS_LEFT ()) THEN NC = 1;
476      1046      4
477      1047      3      END;
478      1048      3
```

```
479 1049 3 [ENT_K_VERT + DIR_FORWARD] :
480 1050 4 BEGIN
481 1051 4
482 1052 4 IF (.EDT$$G_VERT EQL 0) THEN EDT$$SC_CPUNEWCOLPOS ();
483 1053 4
484 1054 5 IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)
485 1055 4 THEN
486 1056 4 SUCCEED = 0
487 1057 4 ELSE
488 1058 5 BEGIN
489 1059 5 EDT$$CS_DWN ();
490 1060 5 EDT$$SC_MATCHCOL ();
491 1061 4 END;
492 1062 4
493 1063 4 EDT$$G_VERT = 1;
494 1064 3 END;
495 1065 3
496 1066 3 [ENT_K_VERT + DIR_BACKWARD] :
497 1067 4 BEGIN
498 1068 4
499 1069 4 IF (.EDT$$G_VERT EQL 0) THEN EDT$$SC_CPUNEWCOLPOS ();
500 1070 4
501 1071 4 SUCCEED = EDT$$CS_UP ();
502 1072 4 EDT$$SC_MATCHCOL ();
503 1073 4 EDT$$G_VERT = 1;
504 1074 3 END;
505 1075 3
506 1076 3 [ENT_K_BW + DIR_FORWARD, ENT_K_BW + DIR_BACKWARD, ENT_K_WORD + DIR_BACKWARD] :
507 1077 4 BEGIN
508 1078 4
509 1079 4 IF EDT$$CS_LEFT () THEN NC = EDT$$FND_BWD (0) + 1 ELSE SUCCEED = 0;
510 1080 4
511 1081 4 EDT$$G_DIR = DIR_BACKWARD;
512 1082 3 END;
513 1083 3
514 1084 3 [ENT_K_BSEN + DIR_FORWARD, ENT_K_BSEN + DIR_BACKWARD, ENT_K_SEN + DIR_BACKWARD] :
515 1085 4 BEGIN
516 1086 4
517 1087 4 IF EDT$$CS_LEFT ()
518 1088 4 THEN
519 1089 5 BEGIN
520 1090 5 LOCAL
521 1091 5 STAT : INITIAL (1);
522 1092 5
523 1093 5 IF (CHRCHAR (.EDT$$A_LN_PTR) EQL ASC_K_CR)
524 1094 6 THEN
525 1095 5 EDT$$CS_LEFT ()
526 1096 5 ELSE
527 1097 5 WHILE ((CHRCHAR (.EDT$$A_LN_PTR) EQL XC' ') AND (.STAT NEQ 0)) DO
528 1098 5 STAT = EDT$$CS_LEFT ();
529 1099 5
530 1100 5 EDT$$FND_BSEN ();
531 1101 5 END
532 1102 5
533 1103 5 ELSE
534 1104 4 SUCCEED = 0;
535 1105 4
```

```
536 1106 4
537 1107 4
538 1108 3
539 1109 3
540 1110 3
541 1111 4
542 1112 4
543 1113 4
544 1114 4
545 1115 4
546 1116 3
547 1117 3
548 1118 3
549 1119 4
550 1120 4
551 1121 5
552 1122 4
553 1123 4
554 1124 4
555 1125 4
556 1126 4
557 1127 4
558 1128 4
559 1129 4
560 1130 3
561 1131 3
562 1132 3
563 1133 4
564 1134 4
565 1135 5
566 1136 4
567 1137 4
568 1138 4
569 1139 4
570 1140 4
571 1141 4
572 1142 4
573 1143 4
574 1144 4
575 1145 3
576 1146 3
577 1147 3
578 1148 4
579 1149 4
580 1150 5
581 1151 4
582 1152 4
583 1153 4
584 1154 5
585 1155 5
586 1156 5
587 1157 4
588 1158 4
589 1159 4
590 1160 3
591 1161 3
592 1162 3

EDT$$G_DIR = DIR_BACKWARD;
END;
[ENT_K_EW + DIR_FORWARD, ENT_K_EW + DIR_BACKWARD, ENT_K_WORD + DIR_FORWARD] :
BEGIN
IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN) THEN SUCCEED = 0 ELSE NC = EDT$$FND_EWD ();
EDT$$G_DIR = DIR_FORWARD;
END;
[ENT_K_SEN + DIR_FORWARD] :
BEGIN
IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)
THEN
SUCCEED = 0
ELSE
WHILE EDT$$CS_RIGHT () DO
IF EDT$$FND_SENGLIM (1) THEN EXITLOOP;
END;
[ENT_K_ESEN + DIR_FORWARD, ENT_K_ESEN + DIR_BACKWARD] :
BEGIN
IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)
THEN
SUCCEED = 0
ELSE
WHILE EDT$$CS_RIGHT () DO
IF EDT$$FND_SENGLIM (0) THEN EXITLOOP;
EDT$$G_DIR = DIR_FORWARD;
END;
[ENT_K_NL + DIR_FORWARD, ENT_K_NL + DIR_BACKWARD, ENT_K_LINE + DIR_FORWARD] :
BEGIN
IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)
THEN
SUCCEED = 0
ELSE
BEGIN
NC = CHSDIFF (.EDT$$A_LN_END, .EDT$$A_LN_PTR) + 1;
SUCCEED = EDT$$CS_DWN ();
END;
EDT$$G_DIR = DIR_FORWARD;
END;
[ENT_K_EL + DIR_FORWARD] :
```

```

593      1163  4      BEGIN
594      1164  4
595      1165  5      IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)
596      1166  4      THEN
597      1167  4          SUCCEED = 0
598      1168  4      ELSE
599      1169  5          BEGIN
600      1170  5              NC = 0;
601      1171  5
602      1172  6          IF CH$PTR_EQL (.EDT$$A_LN_END, .EDT$$A_LN_PTR)
603      1173  5          THEN
604      1174  6              BEGIN
605      1175  6                  EDT$$CS_DWN ();
606      1176  6                  NC = 1;
607      1177  5                  END;
608      1178  5
609      1179  5              NC = .NC + CH$DIFF (.EDT$$A_LN_END, .EDT$$A_LN_PTR);
610      1180  5              EDT$$A_LN_PTR = .EDT$$A_LN_END;
611      1181  4              END;
612      1182  4          END;
613      1183  3
614      1184  3      [ENT_K_EL + DIR_BACKWARD] :
615      1185  3      BEGIN
616      1186  4
617      1187  4      LOCAL
618      1188  4          LEN;
619      1189  4
620      1190  4          LEN = CH$DIFF (.EDT$$A_LN_PTR, CH$PTR (EDT$$ST_LN_BUF));
621      1191  4
622      1192  4      IF (SUCCEED = EDT$$CS_UP ())
623      1193  5      THEN
624      1194  4          BEGIN
625      1195  5              NC = .LEN;
626      1196  5              EDT$$A_LN_PTR = .EDT$$A_LN_END;
627      1197  5              END;
628      1198  4
629      1199  4          END;
630      1200  3
631      1201  3      [ENT_K_BL + DIR_FORWARD, ENT_K_BL + DIR_BACKWARD, ENT_K_LINE + DIR_BACKWARD] :
632      1202  3      BEGIN
633      1203  4
634      1204  4      IF CH$PTR_EQL (.EDT$$A_LN_PTR, CH$PTR (EDT$$ST_LN_BUF))
635      1205  5      THEN
636      1206  4          BEGIN
637      1207  5
638      1208  5              IF (SUCCEED = EDT$$CS_UP ()) THEN NC = .EDT$$G_LN_LEN + 1;
639      1209  5
640      1210  5          END
641      1211  5      ELSE
642      1212  4          BEGIN
643      1213  5              NC = CH$DIFF (.EDT$$A_LN_PTR, CH$PTR (EDT$$ST_LN_BUF));
644      1214  5              EDT$$A_LN_PTR = CH$PTR (EDT$$ST_LN_BUF);
645      1215  5              END;
646      1216  4
647      1217  4      EDT$$G_DIR = DIR_BACKWARD;
648      1218  4      END;
649      1219  3
```

```

650 1220 3
651 1221 3 [ENT_K_PAGE + DIR_FORWARD, ENT_K_EPAGE + DIR_FORWARD, ENT_K_EPAGE + DIR_BACKWARD] :
652 1222 4 BEGIN
653 1223 4 SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [3], DIR_FORWARD, .ENTITY EQL ENT_K_PAGE);
654 1224 4
655 1225 4 IF (.SUCCEED EQL 2) THEN SUCCEED = 0;
656 1226 4
657 1227 4 EDT$$G_DIR = DIR_FORWARD;
658 1228 3 END;
659 1229 3
660 1230 3 [ENT_K_EPAR + DIR_FORWARD, ENT_K_EPAR + DIR_BACKWARD] :
661 1231 4 BEGIN
662 1232 4 SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [2], DIR_FORWARD, 0);
663 1233 4
664 1234 4 IF (.SUCCEED EQL 2) THEN SUCCEED = 0;
665 1235 4
666 1236 4 EDT$$G_DIR = DIR_FORWARD;
667 1237 3 END;
668 1238 3
669 1239 3 [ENT_K_PAGE + DIR_BACKWARD, ENT_K_BPAGE + DIR_BACKWARD, ENT_K_BPAGE + DIR_FORWARD] :
670 1240 4 BEGIN
671 1241 4 SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [3], DIR_BACKWARD, 1);
672 1242 4
673 1243 4 IF (.SUCCEED EQL 2) THEN SUCCEED = 0;
674 1244 4
675 1245 4 EDT$$G_DIR = DIR_BACKWARD;
676 1246 3 END;
677 1247 3
678 1248 3 [ENT_K_BPAR + DIR_BACKWARD, ENT_K_BPAR + DIR_FORWARD] :
679 1249 4 BEGIN
680 1250 4 SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [2], DIR_BACKWARD, 1);
681 1251 4
682 1252 4 IF (.SUCCEED EQL 2) THEN SUCCEED = 0;
683 1253 4
684 1254 4 EDT$$G_DIR = DIR_BACKWARD;
685 1255 3 END;
686 1256 3
687 1257 3 [ENT_K_PAP + DIR_BACKWARD] :
688 1258 4 BEGIN
689 1259 4
690 L 1260 4 %IF SUPPORT_WPS
691 1261 4 %THEN
692 1262 4
693 1263 5 IF (.EDT$$G_PARTYP EQL WSPARA)
694 1264 4 THEN
695 1265 4 SUCCEED = EDT$$FND_WPARA (DIR_BACKWARD)
696 1266 4 ELSE
697 1267 4 %FI
698 1268 4
699 1269 5 BEGIN
700 1270 5 SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [2], DIR_BACKWARD, 1);
701 1271 5
702 1272 5 IF (.SUCCEED EQL 2) THEN SUCCEED = 0;
703 1273 5
704 1274 5 EDT$$G_DIR = DIR_BACKWARD;
705 1275 4 END;
706 1276 4
```

```

: 707      1277  3      END;
: 708      1278  3
: 709      1279  3      [ENT_K_PAR + DIR_FORWARD] :
: 710      1280  4      BEGIN
: 711      1281  4
: 712      1282  4  L  %IF SUPPORT_WPS
: 713      1283  4  %THEN
: 714      1284  4
: 715      1285  5      IF (.EDT$$G_PARTYP EQL WSPARA)
: 716      1286  4      THEN
: 717      1287  4          SUCCEED = EDT$$FND_WPARA (DIR_FORWARD)
: 718      1288  4      ELSE
: 719      1289  4  %FI
: 720      1290  4
: 721      1291  5      BEGIN
: 722      1292  5          SUCCEED = EDT$$FND_ENT (.EDT$$A_US_ENT [2], DIR_FORWARD, 1);
: 723      1293  5
: 724      1294  5          IF (.SUCCEED EQL 2) THEN SUCCEED = 0;
: 725      1295  5
: 726      1296  5          EDT$$G_DIR = DIR_FORWARD;
: 727      1297  4      END;
: 728      1298  4
: 729      1299  3      END;
: 730      1300  3
: 731      1301  3      [ENT_K_BR + DIR_FORWARD, ENT_K_BR + DIR_BACKWARD] :
: 732      1302  4      BEGIN
: 733      1303  4          EDT$$CS_TOP ();
: 734      1304  4          EDT$$G_DIR = DIR_BACKWARD;
: 735      1305  3      END;
: 736      1306  3
: 737      1307  3      [ENT_K_ER + DIR_FORWARD, ENT_K_ER + DIR_BACKWARD] :
: 738      1308  4      BEGIN
: 739      1309  4          EDT$$CS_BOTBUF ();
: 740      1310  4          EDT$$G_DIR = DIR_FORWARD;
: 741      1311  3      END;
: 742      1312  3
: 743      1313  3      [ENT_K_QUOTE + DIR_FORWARD, ENT_K_QUOTE + DIR_BACKWARD] :
: 744      1314  4      BEGIN
: 745      1315  4          SUCCEED = EDT$$STR_SEACMD (EDT$$T_SEA_STR, .EDT$$G_SEA_STRLIN, 1, .EDT$$G_DIR);
: 746      1316  4
: 747      1317  4          IF (.SUCCEED EQL 2) THEN SUCCEED = 0;
: 748      1318  4
: 749      1319  4          IF .SUCCEED
: 750      1320  4      THEN
: 751      1321  4
: 752      1322  5              IF ((.EDT$$G_SEA_BEG EQL 0) AND (.VERB NEQ VERB_K_SSEL))
: 753      1323  4          THEN
: 754      1324  4
: 755      1325  4              DECR I FROM .EDT$$G_SEA_STRLIN - 1 TO 0 DO
: 756      1326  4                  EDT$$CS_RIGHT (I);
: 757      1327  4
: 758      1328  3      END;
: 759      1329  3
: 760      1330  3      [ENT_K_SR + DIR_FORWARD, ENT_K_SR + DIR_BACKWARD] :
: 761      1331  4      BEGIN
: 762      1332  4
: 763      1333  5          IF (.EDT$$A_SEL_BUF EQL .EDT$$A_CUR_BUF)
```



```
764      1334 4      THEN
765      1335 5      BEGIN
766      1336 5
767      L 1337 5 %IF EXTRA_ASSERTS
768      1338 5 %THEN
769      1339 5      ASSERT (.EDT$$A_SEL_POS GEQA EDT$$T_LN_BUF);
770      1340 5 %FI
771      1341 5
772      1342 5 !+
773      1343 5 ! Determine the direction.
774      1344 5 !-
775      1345 5
776      1346 5      (CASE EDT$$SEL_RNGPOS () FROM -1 TO 1 OF
777      1347 5      SET
778      1348 5
779      1349 5      [-1] :      ! Select line is before current line
780      1350 5      EDT$$G_DIR = DIR_BACKWARD;
781      1351 5
782      1352 5      [0] :      ! Select line is current line, check character position
783      1353 5      EDT$$G_DIR = CH$PTR_LSS (.EDT$$A_LN_PTR, .EDT$$A_SEL_POS);
784      1354 5
785      1355 5      [1] :      ! Select line is after current line
786      1356 5      EDT$$G_DIR = DIR_FORWARD;
787      1357 5
788      L 1358 5 %IF EXTRA_ASSERTS
789      1359 5 %THEN
790      1360 5
791      1361 5      [OUTRANGE] :
792      1362 5      ASSERT (0);
793      1363 5 %FI
794      1364 5
795      1365 5      TES:
796      1366 5
797      1367 5 !+
798      1368 5 ! Move up or down until we get to the right line.
799      1369 5 !-
800      1370 5
801      1371 5      WHILE 1 DO
802      1372 5
803      1373 5      CASE EDT$$SEL_RNGPOS () FROM -1 TO 1 OF
804      1374 5      SET
805      1375 5
806      1376 5      [-1] :      ! Select line is before current line
807      1377 5      EDT$$CS_UP ();
808      1378 5
809      1379 5      [0] :      ! Select line is current line
810      1380 5      EXITLOOP;
811      1381 5
812      1382 5      [1] :      ! Select line is after current line
813      1383 5      EDT$$CS_DWN ();
814      1384 5
815      L 1385 5 %IF EXTRA_ASSERTS
816      1386 5 %THEN
817      1387 5
818      1388 5      [OUTRANGE] :
819      1389 5      ASSERT (0);
820      1390 5 %FI
```

```

: 821      1391  5
: 822      1392  5
: 823      1393  5
: 824      1394  5
: 825      1395  5
: 826      1396  5
: 827      1397  5
: 828      1398  5
: 829      1399  5
: 830      1400  5
: 831      1401  5
: 832      1402  5
: 833      1403  5
: 834      1404  5
: 835      1405  5
: 836      1406  5
: 837      1407  5
: 838      1408  5
: 839      1409  5
: 840      1410  5
: 841      1411  4
: 842      1412  5
: 843      1413  5
: 844      1414  5
: 845      1415  5
: 846      1416  5
: 847      1417  5
: 848      1418  5
: 849      1419  6
: 850      1420  5
: 851      1421  5
: 852      1422  5
: 853      1423  5
: 854      1424  5
: 855      1425  5
: 856      1426  6
: 857      1427  6
: 858      1428  5
: 859      1429  5
: 860      1430  6
: 861      1431  5
: 862      1432  6
: 863      1433  6
: 864      1434  6
: 865      1435  6
: 866      1436  6
: 867      1437  6
: 868      1438  6
: 869      1439  5
: 870      1440  5
: 871      1441  5
: 872      1442  5
: 873      1443  6
: 874      1444  6
: 875      1445  7
: 876      1446  6
: 877      1447  6

      TES;
      :- Point to the selected position.
      :-
      EDT$$A_LN_PTR = .EDT$$A_SEL_POS;
L  %IF EXTRA_ASSERTS
      %THEN
      ASSERT ((.EDT$$A_LN_PTR - EDT$$T_LN_BUF) LEQU 255);
      %FI
      :- And turn off the select range, now that we've used it.
      :-
      EDT$$A_SEL_BUF = 0;
      SR = .SR + 1;
      EXITLOOP;
      END
ELSE
      BEGIN
      LOCAL
      CURSOR_MOVES;
      CURSOR_MOVES = 0;
      IF ((.EDT$$G_SEA_BEG EQL 0) AND (.VERB NEQ VERB_K_SEL))
      THEN
      DECR I FROM .EDT$$G_SEA_STRLEN - 1 TO 0 DO
      IF EDT$$CS_LEFT ( ) THEN CURSOR_MOVES = .CURSOR_MOVES + 1;
      IF ((.COUNT EQL 1) AND
      !
      (.EDT$$G_SEA_STRLEN GTR 0) AND EDT$$TST_ONSTR (EDT$$T_SEA_STR, .EDT$$G_SEA_STRLEN))
      THEN
      IF (.EDT$$G_SEA_BEG NEQ 0)
      THEN
      BEGIN
      DECR I FROM .EDT$$G_SEA_STRLEN - 1 TO 0 DO
      EDT$$CS_RIGHT ( );
      EDT$$G_DIR = DIR_FORWARD;
      END
      ELSE
      EDT$$G_DIR = DIR_BACKWARD
      ELSE
      BEGIN
      IF (.EDT$$G_SEA_BEG EQL 0 AND .VERB NEQ VERB_K_SSEL)
      THEN
```

```

: 878      1448 6          DECR I FROM .CURSOR MOVES - 1 TO 0 DO
: 879      1449 6          EDT$$CS_RIGHT (I);
: 880      1450 6
: 881      1451 7          IF ((.VERB EQL VERB_K_CHGC) OR (.VERB EQL VERB_K_CHGU) OR (.VERB EQL VERB_K_CHGL))
: 882      1452 6          THEN
: 883      1453 6
: 884      1454 6          IF (.EDT$$G_DIR EQL DIR_BACKWARD) THEN EDT$$CS_LEFT (I) ELSE EDT$$CS_RIGHT (I)
: 885      1455 6
: 886      1456 6          ELSE
: 887      1457 7          BEGIN
: 888      1458 7          EDT$$MSG BELL (EDT$_NOSELN);
: 889      1459 7          RETURN (0);
: 890      1460 6          END;
: 891      1461 6
: 892      1462 5          END;
: 893      1463 5
: 894      1464 4          END;
: 895      1465 4
: 896      1466 3          END;
: 897      1467 3
: 898      L 1468 3      %IF EXTRA_ASSERTS
: 899      1469 3      %THEN
: 900      1470 3
: 901      1471 3          [OUTRANGE] :
: 902      1472 3          ASSERT (0);
: 903      1473 3      %FI
: 904      1474 3
: 905      1475 3          TES;
: 906      1476 3
: 907      1477 3          IF ( NOT .SUCCEED) THEN EXITLOOP;
: 908      1478 3
: 909      1479 2          END;
: 910      1480 2
: 911      1481 2      !<BLF/PAGE>
```

```

913 1482 2  !+
914 1483 2  !-
915 1484 2  ! If the verb was delete, then save the last entity in the proper
916 1485 2  ! save buffer.
917 1486 2  !-
918 1487 2  IF ((.NC NEQ 0) AND ((.VERB EQL VERB_K_DELETE) OR (.VERB EQL VERB_K_REPLACE)))
919 1488 2  THEN
920 1489 2
921 1490 2  CASE .ENTITY/2 FROM 0 TO ENT_K_EL/2 OF
922 1491 2  SET
923 1492 2
924 1493 2  [ENT_K_CHAR/2] :
925 1494 2  BEGIN
926 1495 2  EDT$$SAV DELTXT (.NC, EDT$$T_DEL_CH, 1);
927 1496 2  EDT$$G_DEL_CHLEN = MIN (.NC, -1);
928 1497 2  END;
929 1498 2
930 1499 2  [ENT_K_WORD/2, ENT_K_BW/2, ENT_K_EW/2] :
931 1500 2  BEGIN
932 1501 2  EDT$$SAV DELTXT (.NC, EDT$$T_DEL_WD, 80);
933 1502 2  EDT$$G_DEL_WDLEN = MIN (.NC, -80);
934 1503 2  END;
935 1504 2
936 1505 2  [ENT_K_LINE/2, ENT_K_NL/2, ENT_K_BL/2, ENT_K_EL/2] :
937 1506 2  BEGIN
938 1507 2  EDT$$SAV DELTXT (.NC, EDT$$T_DEL_LN, 256);
939 1508 2  EDT$$G_DEL_LNLEN = MIN (.NC, -256);
940 1509 2  END;
941 1510 2
942 1511 2  [INRANGE, OUTRANGE] :
943 1512 2  ASSERT (0);
944 1513 2  TES;
945 1514 2
946 1515 2  !+
947 1516 2  ! If the entity was not VERT then turn off the vert flag.
948 1517 2  !-
949 1518 2
950 1519 2  IF (.ENTITY NEQ ENT_K_VERT) THEN EDT$$G_VERT = 0;
951 1520 2
952 1521 2  !+
953 1522 2  ! Calculate the number of lines in the range.
954 1523 2  !-
955 1524 2  SUBLINE (EDT$$A_CUR_BUF [TBCB_CUR_LIN], ORIG_LNO, NUM_LINES);
956 1525 2
957 1526 2  IF ((.NUM_LINES [LN_HI] AND XX'8000') NEQ 0) !
958 1527 2  THEN
959 1528 2  SUBLINE (ORIG_LNC, EDT$$A_CUR_BUF [TBCB_CUR_LIN], NUM_LINES);
960 1529 2
961 1530 2  ADDLINE (NUMBER_ONE, NUM_LINES);
962 1531 2
963 1532 2  !+
964 1533 2  ! Now set up the start and end position blocks, and position to the front of the range.
965 1534 2  !-
966 1535 2  L XIF EXTRA_ASSERTS
967 1536 2  XTHEN
968 1537 2  ASSERT (.START_POS [POS_CHAR_POS] LEQU 255);
969 1538 2  ASSERT ((.EDT$$A_LN_PTR - EDT$$T_LN_BUF) LEQU 255);

```

```
: 970      1539  2 %FI
: 971      1540
: 972      1541      IF (.EDT$$G_DIR EQL DIR_BACKWARD)
: 973      1542      THEN
: 974      1543      BEGIN
: 975      1544      EDT$$CPY_MEM (POS_SIZE, START_POS, END_POS);
: 976      1545
: 977      L 1546      %IF EXTRA_ASSERTS
: 978      1547      %THEN
: 979      1548      ASSERT (.END_POS [POS_CHAR_POS] LEQU 255);
: 980      1549      %FI
: 981      1550
: 982      1551      EDT$$SAV_BUFPOS (START_POS);
: 983      1552      END
: 984      1553      ELSE
: 985      1554      BEGIN
: 986      1555      EDT$$SAV_BUFPOS (END_POS);
: 987      1556
: 988      L 1557      %IF EXTRA_ASSERTS
: 989      1558      %THEN
: 990      1559      ASSERT (.END_POS [POS_CHAR_POS] LEQU 255);
: 991      1560      %FI
: 992      1561
: 993      1562      EDT$$RPOS (START_POS);
: 994      1563      END;
: 995      1564
: 996      L 1565      %IF EXTRA_ASSERTS
: 997      1566      %THEN
: 998      1567      ASSERT (.START_POS [POS_CHAR_POS] LEQU 255);
: 999      1568      ASSERT (.END_POS [POS_CHAR_POS] LEQU 255);
: 1000     1569      %FI
: 1001     1570
: 1002     1571      !<BLF/PAGE>
```

```

: 1004      1572      2  !+
: 1005      1573      2  !-
: 1006      1574      2  !-
: 1007      1575      2  !-
: 1008      1576      2  !-
: 1009      1577      2  !-
: 1010      1578      2  !-
: 1011      1579      2  !-
: 1012      1580      2  !-
: 1013      1581      2  !-
: 1014      1582      2  !-
: 1015      1583      2  !-
: 1016      1584      2  !-
: 1017      1585      2  !-
: 1018      1586      2  !-
: 1019      1587      2  !-
: 1020      1588      2  !-
: 1021      1589      2  !-
: 1022      1590      2  !-
: 1023      1591      2  !-
: 1024      1592      2  !-
: 1025      1593      2  !-
: 1026      1594      2  !-
: 1027      1595      2  !-
: 1028      1596      2  !-
: 1029      1597      2  !-
: 1030      1598      2  !-
: 1031      1599      2  !-
: 1032      1600      2  !-
: 1033      1601      2  !-
: 1034      1602      2  !-
: 1035      1603      2  !-
: 1036      1604      2  !-
: 1037      1605      2  !-
: 1038      1606      2  !-
: 1039      1607      2  !-
: 1040      1608      2  !-
: 1041      1609      2  !-
: 1042      1610      2  !-
: 1043      1611      2  !-
: 1044      1612      2  !-
: 1045      1613      2  !-
: 1046      1614      2  !-
: 1047      1615      2  !-
: 1048      1616      2  !-
: 1049      1617      2  !-
: 1050      1618      2  !-
: 1051      1619      2  !-
: 1052      1620      2  !-
: 1053      1621      2  !-
: 1054      1622      2  !-
: 1055      1623      2  !-
: 1056      1624      2  !-
: 1057      1625      2  !-
: 1058      1626      2  !-
: 1059      1627      2  !-
: 1060      1628      2  !-

      Now, execute the command.

      CASE .VERB FROM VERB_K_MOVE TO VERB_K_APPEND OF
      SET
      [VERB_K_MOVE] :
      BEGIN
      IF (.EDT$$G_DIR EQL DIR_FORWARD) THEN EDT$$RPOS (END_POS);
      END;
      [VERB_K_CHGC] :
      BEGIN
      L 1589 %IF SUPPORT_WPS
      1590 %THEN
      1591 EDT$$G_CAS_TYP = CASE_K_CHGC;
      1592 %FI
      IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
      THEN
      BEGIN
      EDT$$MSG BELL (EDT$_BADRANGE);
      RETURN (0);
      END
      ELSE
      CHANGE_CASE (.NUM_LINES [LN_LO], START_POS, END_POS);
      IF (.SR NEQ 0)
      THEN
      IF (.EDT$$G_DIR EQL DIR_BACKWARD) THEN EDT$$RPOS (END_POS) ELSE EDT$$RPOS (START_POS);
      END;
      [VERB_K_CHGU] :
      BEGIN
      L 1613 %IF SUPPORT_WPS
      1614 %THEN
      1615 EDT$$G_CAS_TYP = CASE_K_CHGU;
      1616
      1617 IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
      1618 THEN
      1619 BEGIN
      1620 EDT$$MSG BELL (EDT$_BADRANGE);
      1621 RETURN (0);
      1622 END
      1623 ELSE
      1624 CHANGE_CASE (.NUM_LINES [LN_LO], START_POS, END_POS);
      1625
      1626 IF (.SR NEQ 0)
      1627 THEN
      1628
```

```

: 1061      1629      3          IF (.EDT$$G_DIR EQL DIR_BACKWARD) THEN EDT$$RPOS (END_POS) ELSE EDT$$RPOS (START_POS);
: 1062      1630      3
: 1063      U 1631      3      %ELSE
: 1064      U 1632      3          0
: 1065      U 1633      3      %FI
: 1066      1634      3
: 1067      1635      2          END;
: 1068      1636      2
: 1069      1637      2      [VERB_K CHGL] :
: 1070      1638      2          BEGIN
: 1071      1639      2
: 1072      L 1640      3      %IF SUPPORT_WPS
: 1073      L 1641      3      %THEN
: 1074      1642      3          EDT$$G_CAS_TYP = CASE_K_CHGL;
: 1075      1643      3
: 1076      1644      3          IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
: 1077      1645      3      THEN
: 1078      1646      4          BEGIN
: 1079      1647      4          EDT$$MSG BELL (EDT$_BADRANGE);
: 1080      1648      4          RETURN (0);
: 1081      1649      4          END
: 1082      1650      3      ELSE
: 1083      1651      3          CHANGE_CASE (.NUM_LINES [LN_LO], START_POS, END_POS);
: 1084      1652      3
: 1085      1653      4          IF (.SR NEQ 0)
: 1086      1654      3      THEN
: 1087      1655      3
: 1088      1656      3          IF (.EDT$$G_DIR EQL DIR_BACKWARD) THEN EDT$$RPOS (END_POS) ELSE EDT$$RPOS (START_POS);
: 1089      1657      3
: 1090      U 1658      3      %ELSE
: 1091      U 1659      3          0
: 1092      U 1660      3      %FI
: 1093      1661      2
: 1094      1662      2          END;
: 1095      1663      2
: 1096      1664      2      [VERB_K SSEL] :          ! search and select command
: 1097      1665      2          BEGIN
: 1098      1666      2
: 1099      L 1667      3      %IF SUPPORT_WPS
: 1100      L 1668      3      %THEN
: 1101      1669      3
: 1102      1670      4          IF (.ENTITY NEQ ENT_K_QUOTE)          ! we are only allowing the search for strings
: 1103      1671      3      THEN
: 1104      1672      4          BEGIN
: 1105      1673      4          EDT$$MSG BELL (EDT$_INVENT);          ! not a string -give error message and
: 1106      1674      4          RETURN (0);          ! get out.
: 1107      1675      3          END;
: 1108      1676      3
: 1109      1677      4          IF (.SUCCEED EQL 1)
: 1110      1678      3      THEN
: 1111      1679      4          BEGIN          ! we were able to find the string
: 1112      1680      4
: 1113      1681      5          IF (.EDT$$G_DIR EQL DIR_FORWARD)
: 1114      1682      4      THEN
: 1115      1683      4          EDT$$RPOS (END_POS)
: 1116      1684      4          ELSE          ! position to the beginning of string
: 1117      1685      4

```

```

: 1118      1686  4      DECR I FROM .EDT$$G_SEA_STRLN - 1 TO 0 DO
: 1119      1687  4      EDT$$CS_RIGHT (I);
: 1120      1688  4
: 1121      1689  5      IF (.EDT$$A_SEL_BUF NEQA 0)
: 1122      1690  4      THEN
: 1123      1691  5      BEGIN
: 1124      1692  5      EDT$$MSG_BELL (EDT$_SELALRACT);
: 1125      1693  5
: 1126      1694  5      IF (.EDT$$G_DIR EQL DIR_FORWARD) THEN EDT$$RPOS (START_POS) ELSE EDT$$RPOS (END_POS);
: 1127      1695  5
: 1128      1696  5      RETURN (0);
: 1129      1697  5      END
: 1130      1698  4      ELSE
: 1131      1699  5      BEGIN
: 1132      1700  5      MOVELINE (EDT$$A_CUR_BUF [TBCB_CUR_LIN], EDT$$L_SEL_LN);
: 1133      1701  5      EDT$$A_SEL_BUF = .EDT$$A_CUR_BUF;
: 1134      1702  5      EDT$$A_SEL_POS = .EDT$$A_LN_PTR;
: 1135      1703  5
: 1136      1704  5      !+ If there was a previous selection during this keystroke, rebuild the screen. The screen
: 1137      1705  5      !- updater cannot handle simultaneous deselection and selection.
: 1138      1706  5
: 1139      1707  5
: 1140      1708  5      IF (.EDT$$A_OLD_SEL NEQA 0) THEN EDT$$G_SCR_REBUILD = 1;
: 1141      1709  5
: 1142      1710  4      END;
: 1143      1711  4
: 1144      1712  4      SUCCEED = EDT$$EXE_CHMCM2 (ENT_K_CHAR, .EDT$$G_SEA_STRLN, VERB_K_MOVE)
: 1145      1713  4      !move across the string with select on
: 1146      1714  3      END;
: 1147      1715  3
: 1148      U 1716  3  %ELSE
: 1149      U 1717  3      0
: 1150      1718  3  %FI
: 1151      1719  3
: 1152      1720  2      END;
: 1153      1721  2
: 1154      1722  2  [VERB_K_TADJ] :
: 1155      1723  3      BEGIN
: 1156      1724  3
: 1157      1725  4      IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
: 1158      1726  3      THEN
: 1159      1727  4      BEGIN
: 1160      1728  4      EDT$$MSG_BELL (EDT$_BADRANGE);
: 1161      1729  4      RETURN (0);
: 1162      1730  4      END
: 1163      1731  3      ELSE
: 1164      1732  3
: 1165      1733  3      INCR I FROM 1 TO (.NUM_LINES [LN_LO] - (.END_POS [POS_CHAR_POS] EQL 0)) DO
: 1166      1734  4      BEGIN
: 1167      1735  4      EDT$$TADJ_CMD ();
: 1168      1736  4      EDT$$CS_DWN ();
: 1169      1737  3      END;
: 1170      1738  3
: 1171      1739  2      END;
: 1172      1740  2
: 1173      1741  3  [VERB_K_DELETE, VERB_K_REPLACE] :
: 1174      1742  3      BEGIN

```



```

: 1175      1743  3      EDT$$A_ALT_BUF = 0;
: 1176      1744  3
: 1177      1745  4      IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
: 1178      1746  3      THEN
: 1179      1747  4          BEGIN
: 1180      1748  4          EDT$$MSG BELL (EDT$_BADRANGE);
: 1181      1749  4          RETURN (0);
: 1182      1750  4          END
: 1183      1751  3      ELSE
: 1184      1752  4          BEGIN
: 1185      1753  4          EDT$$DEL_TXTLN (.NUM_LINES [LN_LO], CH$PTR (EDT$$T_LN_BUF, .END_POS [POS_CHAR_POS]));
: 1186      1754  4
: 1187      1755  4          IF (.VERB EQL VERB_K_REPLACE) THEN SUCCEED = EDT$$INS_MOD ();
: 1188      1756  4
: 1189      1757  3          END;
: 1190      1758  3      END;
: 1191      1759  2
: 1192      1760  2      [VERB K FILL] :
: 1193      1761  2      BEGIN
: 1194      1762  3
: 1195      1763  3      IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
: 1196      1764  4      THEN
: 1197      1765  3          BEGIN
: 1198      1766  4          EDT$$MSG BELL (EDT$_BADRANGE);
: 1199      1767  4          RETURN (0);
: 1200      1768  4          END
: 1201      1769  4      ELSE
: 1202      1770  3          BEGIN
: 1203      1771  4          EDT$$RPL_CHGDLN ();
: 1204      1772  4          EDT$$FILC_TXT (.NUM_LINES [LN_LO] - (.END_POS [POS_CHAR_POS] EQL 0));
: 1205      1773  4          EDT$$GET_TXTLN ();
: 1206      1774  4          END;
: 1207      1775  3      END;
: 1208      1776  3
: 1209      1777  2      [VERB K CUT, VERB_K_APPEND] :
: 1210      1778  2      BEGIN
: 1211      1779  2      LOCAL
: 1212      1780  3          SAVE_BUF;
: 1213      1781  3
: 1214      1782  3          IF (.EDT$$A_ALT_BUF EQL 0) THEN RETURN (0);
: 1215      1783  3
: 1216      1784  3          IF (.EDT$$A_ALT_BUF EQL .EDT$$A_CUR_BUF)
: 1217      1785  3          THEN
: 1218      1786  3              BEGIN
: 1219      1787  4              EDT$$MSG BELL (EDT$_ATTCUTAPP);
: 1220      1788  3              RETURN (0);
: 1221      1789  4              END;
: 1222      1790  4
: 1223      1791  4          IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
: 1224      1792  3          THEN
: 1225      1793  3              BEGIN
: 1226      1794  4              EDT$$MSG BELL (EDT$_BADRANGE);
: 1227      1795  3              RETURN (0);
: 1228      1796  4              END
: 1229      1797  4
: 1230      1798  4
: 1231      1799  4
```

```

: 1232      1800   3      ELSE
: 1233      1801   4      BEGIN
: 1234      1802   4      SAVE_BUF = .EDT$$A_CUR_BUF;
: 1235      1803   4      EDT$$A_CUR_BUF = .EDT$$A_ALT_BUF;
: 1236      1804   4
: 1237      1805   4      IF (.VERB EQL VERB_K_APPEND) THEN EDT$$WF_BOT () ELSE EDT$$WF_CLRBUF ();
: 1238      1806   4
: 1239      1807   4      EDT$$START_INS ();
: 1240      1808   4      EDT$$A_CUR_BUF = .SAVE_BUF;
: 1241      1809   4      EDT$$DEL_TXTLN (.NUM_LINES [LN_LO], CH$PTR (EDT$$T_LN_BUF, .END_POS [POS_CHAR_POS]));
: 1242      1810   4      EDT$$RPL_CHGDLN ();
: 1243      1811   4      EDT$$A_CUR_BUF [TBCB_CHAR_POS] = CH$DIFF (.EDT$$A_LN_PTR, CH$PTR (EDT$$T_LN_BUF));
: 1244      1812   4      EDT$$A_CUR_BUF = .EDT$$A_ALT_BUF;
: 1245      1813   4      EDT$$RD_CURLN ();
: 1246      1814   4      EDT$$END_INS ();
: 1247      1815   4      !+
: 1248      1816   4      !- If this is an append, combine first line appended with previous line.
: 1249      1817   4      !-
: 1250      1818   4
: 1251      1819   5      IF (.VERB EQL VERB_K_APPEND)
: 1252      1820   4      THEN
: 1253      1821   5      BEGIN
: 1254      1822   5
: 1255      1823   5      DECR I FROM .NUM_LINES [LN_LO] - 1 TO 0 DO
: 1256      1824   5      EDT$$RD_PRVLN ();
: 1257      1825   5
: 1258      1826   5      EDT$$GET_TXTLN ();
: 1259      1827   5
: 1260      1828   5      IF EDT$$RD_PRVLN ()
: 1261      1829   5      THEN
: 1262      1830   5
: 1263      1831   6      IF ((.EDT$$G_LN_LEN + .EDT$$A_WK_LN [LIN_LENGTH]) LEQ 255)
: 1264      1832   5      THEN
: 1265      1833   6      BEGIN
: 1266      1834   6      EDT$$CPY_MEM (.EDT$$G_LN_LEN, CH$PTR (EDT$$T_LN_BUF, !
: 1267      1835   6      CH$PTR (EDT$$T_LN_BUF, .EDT$$A_WK_LN [LIN_LENGTH]));
: 1268      1836   6      EDT$$CPY_MEM (.EDT$$A_WK_LN [LIN_LENGTH], EDT$$A_WK_LN [LIN_TEXT],
: 1269      1837   6      CH$PTR (EDT$$T_LN_BUF));
: 1270      1838   6      EDT$$G_LN_LEN = .EDT$$G_LN_LEN + .EDT$$A_WK_LN [LIN_LENGTH];
: 1271      1839   6      EDT$$DEL_CURLN ();
: 1272      1840   6      EDT$$RPL_LN (EDT$$T_LN_BUF, .EDT$$G_LN_LEN);
: 1273      1841   5      END;
: 1274      1842   5
: 1275      1843   4      END;
: 1276      1844   4
: 1277      1845   4      EDT$$A_CUR_BUF = .SAVE_BUF;
: 1278      1846   4      EDT$$GET_TXTLN ();
: 1279      1847   3      END;
: 1280      1848   3
: 1281      1849   2      END;
: 1282      1850   2
: 1283      1851   2      %IF EXTRA_ASSERTS
: 1284      1852   2      %THEN
: 1285      1853   2
: 1286      1854   2      [INRANGE, OVRANGE] :
: 1287      1855   2      ASSERT (0);
: 1288      1856   2      %FI

```

: 1289
: 1290
: 1291
: 1292
: 1293
: 1294
: 1295
: 1296
: 1297
: 1298
: 1299
: 1300
: 1301
: 1302
: 1303
: 1304

1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872

2
2
3
2
2
2
3
2
2
2
2
2
2
2
2
1

```
TES;  
IF (.SUCCEED EQL 0)  
THEN  
  EDT$$MSG_BELL (  
    IF (.ENTITY EQL ENT_K_QUOTE)  
    THEN  
      EDT$_STRNOTFND  
    ELSE  
      IF (.EDT$$G_DIR EQL DIR_BACKWARD) THEN EDT$_TOPOFBUF ELSE EDT$_BOTOFBUF);  
RETURN (.SUCCEED);  
END;
```

! of routine EDT\$\$EXE_CHMCM2

.TITLE EDT\$CHMEXVERB EDT\$CHMEXVERB - execute certain c
change-mode com

.IDENT \V04-000\

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

4B 43 41 42 0000 P.AAA:
56 44 41 00004 P.AAB:

.ASCII \BACK\
.ASCII \ADV\

:

- .EXTRN EDT\$\$FND_BSEN, EDT\$\$FND_BWD
- .EXTRN EDT\$\$DEL_TXTLN, EDT\$\$MSG_BELL
- .EXTRN EDT\$\$CHK_CC, EDT\$\$STADJ_CMD
- .EXTRN EDT\$\$FND_EWD, EDT\$\$FND_ENT
- .EXTRN EDT\$\$STR_SEACMD
- .EXTRN EDT\$\$FILE_TXT, EDT\$\$INS_MOD
- .EXTRN EDT\$\$CS_BOTBUF, EDT\$\$RPC_CHGDLN
- .EXTRN EDT\$\$CS_DWN, EDT\$\$GET_TXTLN
- .EXTRN EDT\$\$CS_LEFT, EDT\$\$CS_RIGHT
- .EXTRN EDT\$\$CS_TOP, EDT\$\$CS_OP
- .EXTRN EDT\$\$TST_ONSTR, EDT\$\$RPOS
- .EXTRN EDT\$\$SAV_BUFPOS
- .EXTRN EDT\$\$SAV_DELTXT
- .EXTRN EDT\$\$SC_MATCHCOL
- .EXTRN EDT\$\$SC_CPUNEWCOLPOS
- .EXTRN EDT\$\$SEC_RNGPOS
- .EXTRN EDT\$\$FND_SENDLIM
- .EXTRN EDT\$\$STR_CMP, EDT\$\$SET_SEASTR
- .EXTRN EDT\$\$WF_BOT, EDT\$\$WF_C[RBUF
- .EXTRN EDT\$\$END_INS, EDT\$\$DEL_CURLN
- .EXTRN EDT\$\$RPL_LN, EDT\$\$RD_PV_LN
- .EXTRN EDT\$\$RD_CURLN, EDT\$\$FND_WPARA
- .EXTRN EDT\$\$START_INS, EDT\$\$G_CAS_TYP
- .EXTRN EDT\$\$T_DEL_CH, EDT\$\$G_DEL_CHLEN
- .EXTRN EDT\$\$T_DEL_LN, EDT\$\$G_DEL_LNLEN
- .EXTRN EDT\$\$G_DIR, EDT\$\$T_DEC_WD
- .EXTRN EDT\$\$G_DEL_WDLEN
- .EXTRN EDT\$\$A_CMD_BUF, EDT\$\$A_CMD_END
- .EXTRN EDT\$\$A_SEA_STR, EDT\$\$A_SEL_BUF
- .EXTRN EDT\$\$A_OLD_SEL, EDT\$\$G_SCR_REBUILD

				OFFC 00000	.EXTRN	EDT\$\$L_SEL_LN, EDT\$\$A_SEL_POS	
					.EXTRN	EDT\$\$G_SEA_LEN, EDT\$\$G_VERT	
					.EXTRN	EDT\$\$A_ALT_BUF, EDT\$\$A_US_ENT	
					.EXTRN	EDT\$\$G_SEA_BEG, EDT\$\$A_CUR_BUF	
					.EXTRN	EDT\$\$G_TI_TYP, EDT\$\$T_CN_BUF	
					.EXTRN	EDT\$\$A_LN_END, EDT\$\$G_LN_LEN	
					.EXTRN	EDT\$\$A_LN_PTR, EDT\$\$G_SEA_STRLEN	
					.EXTRN	EDT\$\$T_SEA_STR, EDT\$\$G_PARTYP	
					.EXTRN	EDT\$\$A_WK_LN, EDT\$\$Z_EOB_LN	
					.EXTRN	EDT\$\$G_CC_DONE, EDT\$\$L_LN00	
					.EXTRN	EDT\$_NOSECRAN, EDT\$_SEALRACT	
					.EXTRN	EDT\$_INVENT, EDT\$_ATTICUTAPP	
					.EXTRN	EDT\$_STRNOTFND, EDT\$_TOPOFBUF	
					.EXTRN	EDT\$_BOTOFBUF, EDT\$_INVSTR	
					.EXTRN	EDT\$_BADRANGE, EDT\$\$INTER_ERR	
					.ENTRY	EDT\$\$EXE_CHMCM2, Save R2,R3,R4,R5,R6,R7,-	0709
						R8,R9,R10,R11	
					SUBL2	#48, SP	
					MOVL	#1, SUCCEED	0891
					CLRL	SR	0892
					CLRL	NC	0893
					MOVL	ENTITY, R9	0898
					CLRL	-(SP)	
					CMPL	R9, #41	
					BNEQ	3\$	
					INCL	(SP)	
					PUSHL	EDT\$\$G_SEA_LEN	0902
					PUSHL	EDT\$\$A_SEA_STR	
					CALLS	#2, EDT\$\$SET_SEASTR	
					BLBS	RO, 1\$	
					CLRL	SUCCEED	0905
					PUSHL	#EDT\$_INVSTR	0906
					BRW	192\$	
					MOVL	EDT\$\$A_CMD_BUF, RO	0916
					CMPB	(RO), #32	
					BNEQ	2\$	
					ADDL2	#3, RO	
					CMPL	RO, EDT\$\$A_CMD_END	
					BGTRU	2\$	
					INCL	EDT\$\$A_CMD_BUF	0917
					BRB	1\$	
					MOVL	EDT\$\$A_CMD_BUF, RO	0919
					MOVAB	3(RO), R1	
					CMPL	R1, EDT\$\$A_CMD_END	
					BGTRU	5\$	
					PUSHL	#3	0923
					PUSHL	#4	
					PUSHAB	P.AAA	
					PUSHL	RO	
					CALLS	#4, EDT\$\$STR_CMP	
					BLBC	RO, 4\$	
					CLRL	EDT\$\$G_DIR	0925
					BRB	5\$	
					PUSHL	#3	0928
					PUSHL	#3	
					PUSHAB	P.AAB	

	00000000G	00		7E D4 00147	CLRL	-(SP)		0976
		57		01 FB 00149	CALLS	#1, EDT\$\$FND_WPARA		
				50 D0 00150	MOVL	R0, SUCCEED		
				20 11 00153	BRB	17\$		
				01 DD 00155	14\$: PUSHL	#1		0981
				7E D4 00157	CLRL	-(SP)		
	00000000G	00	00000000G	00 DD 00159	PUSHL	EDT\$\$A_US_ENT+8		
		57		03 FB 0015F	15\$: CALLS	#3, EDT\$\$FND_ENT		
		02		50 D0 00166	MOVL	R0, SUCCEED		
				57 D1 00169	CMPL	SUCCEED, #2		0983
				04 12 0016C	BNEQ	16\$		
				57 D4 0016E	CLRL	SUCCEED		
				03 11 00170	BRB	17\$		
		57		01 D0 00172	16\$: MOVL	#1, SUCCEED		
			24	AE 9F 00175	17\$: PUSHAB	START_POS		1006
	00000000G	00		01 FB 00178	CALLS	#1, EDT\$\$SAV_BUFPOS		
				5A D4 0017F	CLRL	R10		1010
	00FF	8F	30	AE B1 00181	CMPL	START_POS+12, #255		
				02 1A 00187	BGTRU	18\$		
				5A D6 00189	INCL	R10		
		07		5A E8 0018B	18\$: BLBS	R10, 19\$		
	00000000G	00		00 FB 0018E	CALLS	#0, EDT\$\$INTER_ERR		
		50	00000000G	00 D0 00195	19\$: MOVL	EDT\$\$A_CUR_BUF, R0		1013
OC	AE	06	A0	06 28 0019C	MOV C3	#6, 6(R0), -ORIG_LNO		
				54 D4 001A2	CLRL	I		1018
				0526 31 001A4	BRW	117\$		
	00000000G	00		00 FB 001A7	20\$: CALLS	#0, EDT\$\$CHK_CC		1021
				50 D5 001AE	TSTL	R0		
				0A 13 001B0	BEQL	21\$		
	00000000G	00		01 D0 001B2	MOVL	#1, EDT\$\$G_CC_DONE		1024
				0518 31 001B9	BRW	118\$		1023
		53	00000000G	00 D0 001BC	21\$: MOVL	EDT\$\$G_DIR, R3		1032
		59		53 C1 001C3	ADDL3	R3, R9, R0		
		01		50 CF 001C7	CASEL	R0, #1, #43		
0139	00D5	0061	006A	001CB	22\$: .WORD	24\$-22\$, -		
0139	0139	00D5	00D5	001D3		23\$-22\$, -		
0251	0251	01A8	0251	001D8		35\$-22\$, -		
007F	00AB	01A8	01A8	001E3		42\$-22\$, -		
0155	00EE	01D9	0221	001EB		35\$-22\$, -		
0180	0180	00EE	00EE	001F3		35\$-22\$, -		
02C2	02C2	02DF	02B6	001FB		42\$-22\$, -		
0293	02AA	02A6	02A6	00203		42\$-22\$, -		
0293	0293	02AA	02AA	00208		60\$-22\$, -		
031B	031B	0311	0311	00213		47\$-22\$, -		
036F	036F	0325	0325	0021B		60\$-22\$, -		
						60\$-22\$, -		
						47\$-22\$, -		
						47\$-22\$, -		
						30\$-22\$, -		
						27\$-22\$, -		
						57\$-22\$, -		
						52\$-22\$, -		
						36\$-22\$, -		
						43\$-22\$, -		
						36\$-22\$, -		
						36\$-22\$, -		
						45\$-22\$, -		

					45\$-22\$,-		
					66\$-22\$,-		
					71\$-22\$,-		
					67\$-22\$,-		
					67\$-22\$,-		
					64\$-22\$,-		
					64\$-22\$,-		
					65\$-22\$,-		
					62\$-22\$,-		
					65\$-22\$,-		
					65\$-22\$,-		
					62\$-22\$,-		
					62\$-22\$,-		
					76\$-22\$,-		
					76\$-22\$,-		
					78\$-22\$,-		
					78\$-22\$,-		
					80\$-22\$,-		
					80\$-22\$,-		
					86\$-22\$,-		
					86\$-22\$,-		
00000000G	00	00	FB 00223	CALLS	#0, EDT\$\$INTER_ERR		1472
		71	11 0022A	BRB	34\$		1032
00000000G	00	00	FB 0022C 23\$:	CALLS	#0, EDT\$\$CS_RIGHT		1038
		07	11 00233	BRB	25\$		
00000000G	00	00	FB 00235 24\$:	CALLS	#0, EDT\$\$CS_LEFT		1045
	57	50	D0 0023C 25\$:	MOVL	R0, SUCCEED		
	03	57	E8 0023F	BLBS	SUCCEED, 26\$		
		048F	31 00242	BRW	118\$		
	56	01	D0 00245 26\$:	MOVL	#1, NC		
		53	11 00248	BRB	34\$		1032
		00000000G	00 D5 0024A 27\$:	TSTL	EDT\$\$G_VERT		1052
		07	12 00250	BNEQ	28\$		
00000000G	00	00	FB 00252	CALLS	#0, EDT\$\$SC_CPUNEWCOLPOS		
	50	00000000G	00 9E 00259 28\$:	MOVAB	EDT\$\$Z_EOB_CN, R0		1054
	50	00000000G	00 D1 00260	CMPL	EDT\$\$A_WK_CN, R0		
		04	12 00267	BNEQ	29\$		
		57	D4 00269	CLRL	SUCCEED		1056
		29	11 0026B	BRB	33\$		
00000000G	00	00	FB 0026D 29\$:	CALLS	#0, EDT\$\$CS_DWN		1059
		19	11 00274	BRB	32\$		1060
		00000000G	00 D5 00276 30\$:	TSTL	EDT\$\$G_VERT		1069
		07	12 0027C	BNEQ	31\$		
00000000G	00	00	FB 0027E	CALLS	#0, EDT\$\$SC_CPUNEWCOLPOS		
00000000G	00	00	FB 00285 31\$:	CALLS	#0, EDT\$\$CS_UP		1071
	57	50	D0 0028C	MOVL	R0, SUCCEED		
00000000G	00	00	FB 0028F 32\$:	CALLS	#0, EDT\$\$SC_MATCHCOL		1072
00000000G	00	01	D0 00296 33\$:	MOVL	#1, EDT\$\$G_VERT		1073
		042A	31 0029D 34\$:	BRW	116\$		1032
00000000G	00	00	FB 002A0 35\$:	CALLS	#0, EDT\$\$CS_LEFT		1079
	16	50	E9 002A7	BLBC	R0, 37\$		
		7E	D4 002AA	CLRL	-(SP)		
00000000G	00	01	FB 002AC	CALLS	#1, EDT\$\$FND_BWD		
	56	01	9E 002B3	MOVAB	1(R0), NC		
		48	11 002B7	BRB	41\$		
00000000G	00	00	FB 002B9 36\$:	CALLS	#0, EDT\$\$CS_LEFT		1087
	03	50	E8 002C0 37\$:	BLBS	R0, 38\$		

		01E0	31	002C3	BRW	69\$		
52		01	D0	002C6	38\$:	MOVL	#1, STAT	1089
50	00000000G	00	D0	002C9		MOVL	EDT\$\$A_LN_PTR, RO	1094
0D		60	91	002D0		CMPB	(RO), #13	
		09	12	002D3		BNEQ	39\$	
00000000G	00	00	FB	002D5		CALLS	#0, EDT\$\$CS_LEFT	1096
		1C	11	002DC		BRB	40\$	
50	00000000G	00	D0	002DE	39\$:	MOVL	EDT\$\$A_LN_PTR, RO	1099
20		60	91	002E5		CMPB	(RO), #32	
		10	12	002E8		BNEQ	40\$	
		52	D5	002EA		TSTL	STAT	
		0C	13	002EC		BEQL	40\$	
00000000G	00	00	FB	002EE		CALLS	#0, EDT\$\$CS_LEFT	1100
		52	D0	002F5		MOVL	RO, STAT	
		E4	11	002F8		BRB	39\$	
00000000G	00	00	FB	002FA	40\$:	CALLS	#0, EDT\$\$FND_BSEN	1102
		0373	31	00301	41\$:	BRW	108\$	1087
50	00000000G	00	9E	00304	42\$:	MOVAB	EDT\$\$Z_EOB_LN, RO	1113
50	00000000G	00	D1	0030B		CMPB	EDT\$\$A_WK_CN, RO	
		6F	13	00312		BEQL	48\$	
00000000G	00	00	FB	00314		CALLS	#0, EDT\$\$FND_EWD	
		56	D0	0031B		MOVL	RO, NC	
		65	11	0031E		BRB	49\$	1115
50	00000000G	00	9E	00320	43\$:	MOVAB	EDT\$\$Z_EOB_LN, RO	1121
50	00000000G	00	D1	00327		CMPB	EDT\$\$A_WK_CN, RO	
		03	12	0032E		BNEQ	44\$	
		0081	31	00330		BRW	53\$	
00000000G	00	00	FB	00333	44\$:	CALLS	#0, EDT\$\$CS_RIGHT	1126
		79	E9	0033A		BLBC	RO, 54\$	
		01	DD	0033D		PUSHL	#1	1128
00000000G	00	01	FB	0033F		CALLS	#1, EDT\$\$FND_SENDLIM	
		EA	E9	00346		BLBC	RO, 44\$	
		6B	11	00349		BRB	54\$	
50	00000000G	00	9E	0034B	45\$:	MOVAB	EDT\$\$Z_EOB_LN, RO	1135
50	00000000G	00	D1	00352		CMPB	EDT\$\$A_WK_CN, RO	
		28	13	00359		BEQL	48\$	
00000000G	00	00	FB	0035B	46\$:	CALLS	#0, EDT\$\$CS_RIGHT	1140
		3C	E9	00362		BLBC	RO, 51\$	
		7E	D4	00365		CLRL	-(SP)	1142
00000000G	00	01	FB	00367		CALLS	#1, EDT\$\$FND_SENDLIM	
		EA	E9	0036E		BLBC	RO, 46\$	
		2E	11	00371		BRB	51\$	1144
50	00000000G	00	9E	00373	47\$:	MOVAB	EDT\$\$Z_EOB_LN, RO	1150
50	00000000G	00	D1	0037A		CMPB	EDT\$\$A_WK_CN, RO	
		04	12	00381		BNEQ	50\$	
		57	D4	00383	48\$:	CLRL	SUCCEED	1152
		1A	11	00385	49\$:	BRB	51\$	
50	00000000G	00	C3	00387	50\$:	SUBL3	EDT\$\$A_LN_PTR, EDT\$\$A_LN_END, RO	1155
		56	A0	00393		MOVAB	1(RO), NC	
00000000G	00	00	FB	00397		CALLS	#0, EDT\$\$CS_DWN	1156
		57	D0	0039E		MOVL	RO, SUCCEED	
		02CA	31	003A1	51\$:	BRW	107\$	1159
50	00000000G	00	9E	003A4	52\$:	MOVAB	EDT\$\$Z_EOB_LN, RO	1165
50	00000000G	00	D1	003AB		CMPB	EDT\$\$A_WK_CN, RO	
		04	12	003B2		BNEQ	55\$	
		57	D4	003B4	53\$:	CLRL	SUCCEED	1167
		61	11	003B6	54\$:	BRB	59\$	

			56	D4	003B8	55\$:	CLRL	NC	1170		
	00000000G	00	00	D1	003BA		CMPL	EDT\$\$A_LN_END, EDT\$\$A_LN_PTR	1172		
			0A	12	003C5		BNEQ	56\$			
	00000000G	00	00	FB	003C7		CALLS	#0, EDT\$\$CS_DWN	1175		
		56	01	D0	003CE		MOVL	#1, NC	1176		
		50	00000000G	00	D0	003D1	56\$:	MOVL	EDT\$\$A_LN_END, R0	1179	
51		50	00000000G	00	C3	003D8		SUBL3	EDT\$\$A_LN_PTR, R0, R1		
		56		51	C0	003E0		ADDL2	R1, NC		
	00000000G	00		50	D0	003E3		MOVL	R0, EDT\$\$A_LN_PTR	1180	
				2D	11	003EA		BRB	59\$	1032	
		50	00000000G	00	9E	003EC	57\$:	MOVAB	EDT\$\$T_LN_BUF, R0	1191	
52	00000000G	00		50	C3	003F3		SUBL3	R0, EDT\$\$A_LN_PTR, LEN		
	00000000G	00		00	FB	003FB		CALLS	#0, EDT\$\$CS_UP	1193	
		57		50	D0	00402		MOVL	R0, SUCCEED		
		03		57	E8	00405		BLBS	SUCCEED, 58\$		
			02C9	31	00408		BRW	118\$			
		56		52	D0	0040B	58\$:	MOVL	LEN, NC	1196	
	00000000G	00	00000000G	00	D0	0040E		MOVL	EDT\$\$A_LN_END, EDT\$\$A_LN_PTR	1197	
				02AE	31	00419	59\$:	BRW	116\$	1032	
		52	00000000G	00	D0	0041C	60\$:	MOVL	EDT\$\$A_LN_PTR, R2	1205	
		50	00000000G	00	9E	00423		MOVAB	EDT\$\$T_LN_BUF, R0		
		50		52	D1	0042A		CMPL	R2, R0		
				17	12	0042D		BNEQ	61\$		
	00000000G	00		00	FB	0042F		CALLS	#0, EDT\$\$CS_UP	1209	
		57		50	D0	00436		MOVL	R0, SUCCEED		
		6C		57	E9	00439		BLBC	SUCCEED, 70\$		
56	00000000G	00		01	C1	0043C		ADDL3	#1, EDT\$\$G_LN_LEN, NC		
				62	11	00444		BRB	70\$	1205	
		50	00000000G	00	9E	00446	61\$:	MOVAB	EDT\$\$T_LN_BUF, R0	1214	
		52		50	C3	0044D		SUBL3	R0, R2, NC		
56	00000000G	00	00000000G	00	9E	00451		MOVAB	EDT\$\$T_LN_BUF, EDT\$\$A_LN_PTR	1215	
				4A	11	0045C		BRB	70\$	1218	
				7E	D4	0045E	62\$:	CLRL	-(SP)	1223	
		1F		59	D1	00460		CMPL	R9, #31		
				02	12	00463		BNEQ	63\$		
				6E	D6	00465		INCL	(SP)		
				01	DD	00467	63\$:	PUSHL	#1		
			00000000G	00	DD	00469		PUSHL	EDT\$\$A_US_ENT+12		
				59	11	0046F		BRB	75\$		
				7E	D4	00471	64\$:	CLRL	-(SP)	1232	
				4D	11	00473		BRB	74\$		
				01	DD	00475	65\$:	PUSHL	#1	1241	
				7E	D4	00477		CLRL	-(SP)		
			00000000G	00	DD	00479		PUSHL	EDT\$\$A_US_ENT+12		
				16	11	0047F		BRB	68\$		
				00000000G	00	D5	00481	66\$:	TSTL	EDT\$\$G_PARTYP	1263
				04	12	00487		BNEQ	67\$		
				7E	D4	00489		CLRL	-(SP)	1265	
				27	11	0048B		BRB	72\$		
				01	DD	0048D	67\$:	PUSHL	#1	1270	
				7E	D4	0048F		CLRL	-(SP)		
			00000000G	00	DD	00491		PUSHL	EDT\$\$A_US_ENT+8		
	00000000G	00		03	FB	00497	68\$:	CALLS	#3, EDT\$\$FND_ENT		
		57		50	D0	0049E		MOVL	R0, SUCCEED		
		02		57	D1	004A1		CMPL	SUCCEED, #2	1272	
				3D	12	004A4		BNEQ	77\$		
				57	D4	004A6	69\$:	CLRL	SUCCEED		

			39	11	004A8	70\$:	BRB	77\$:	1274
		00000000G	00	D5	004AA	71\$:	TSTL	EDT\$\$G_PARTYP	:	1285
			0E	12	004B0		BNEQ	73\$:	
			01	DD	004B2		PUSHL	#1	:	1287
00000000G	00		01	FB	004B4	72\$:	CALLS	#1, EDT\$\$FND_WPARA	:	
	57		50	D0	004BB		MOVL	R0, SUCCEED	:	
			77	11	004BE		BRB	85\$:	
			01	DD	004C0	73\$:	PUSHL	#1	:	1292
			01	DD	004C2	74\$:	PUSHL	#1	:	
00000000G	00	00000000G	00	DD	004C4		PUSHL	EDT\$\$A_US_ENT+8	:	
	57		03	FB	004CA	75\$:	CALLS	#3, EDT\$\$FND_ENT	:	
	02		50	D0	004D1		MOVI	R0, SUCCEED	:	
			57	D1	004D4		CMPI	SUCCEED, #2	:	1294
			14	12	004D7		BNEQ	79\$:	
00000000G	00		FEA7	31	004D9		BRW	48\$:	
			00	FB	004DC	76\$:	CALLS	#0, EDT\$\$CS_TOP	:	1303
			0191	31	004E3	77\$:	BRW	108\$:	1304
00000000G	00		00	FB	004E6	78\$:	CALLS	#0, EDT\$\$CS_POTBUF	:	1309
			017E	31	004ED	79\$:	BRW	107\$:	1310
			53	DD	004F0	80\$:	PUSHL	R3	:	1315
			01	DD	004F2		PUSHL	#1	:	
		00000000G	00	DD	004F4		PUSHL	EDT\$\$G_SEA_STRLN	:	
00000000G	00	00000000G	00	9F	004FA		PUSHAB	EDT\$\$T_SEA_STR	:	
	57		04	FB	00500		CALLS	#4, EDT\$\$STR_SEACMD	:	
	02		50	D0	00507		MOVL	R0, SUCCEED	:	
			57	D1	0050A		CMPL	SUCCEED, #2	:	1317
			02	12	0050D		BNEQ	81\$:	
			57	D4	0050F		CLRL	SUCCEED	:	
	03		57	E8	00511	81\$:	BLBS	SUCCEED, 82\$:	1319
			018D	31	00514		BRW	118\$:	
		00000000G	00	D5	00517	82\$:	TSTL	EDT\$\$G_SEA_BEG	:	1322
			18	12	0051D		BNEQ	85\$:	
	06		58	D1	0051F		CMPL	R8, #6	:	
			13	13	00522		BEQL	85\$:	
	52	00000000G	00	D0	00524		MOVL	EDT\$\$G_SEA_STRLN, I	:	1325
			07	11	0052B		BRB	84\$:	
00000000G	00		00	FB	0052D	83\$:	CALLS	#0, EDT\$\$CS_RIGHT	:	1326
	F6		52	F4	00534	84\$:	SOBGEQ	I, 83\$:	
			0190	31	00537	85\$:	BRW	116\$:	1032
00000000G	00	00000000G	00	D1	0053A	86\$:	CMPL	EDT\$\$A_SEL_BUF, EDT\$\$A_CUR_BUF	:	1333
			03	13	00545		BEQL	87\$:	
			00BE	31	00547		BRW	101\$:	
	50	00000000G	00	9E	0054A	87\$:	MOVAB	EDT\$\$T_LN_BUF, R0	:	1339
	50	00000000G	00	D1	00551		CMPL	EDT\$\$A_SEC_POS, R0	:	
			07	1E	00558		BGEQU	88\$:	
00000000G	00		00	FB	0055A		CALLS	#0, EDT\$\$INTER_ERR	:	
00000000G	00		00	FB	00561	88\$:	CALLS	#0, EDT\$\$SEL_RNGPOS	:	1346
02	FFFFFFFF	8F	50	CF	00568		CASEL	R0, #-1, #2	:	
002A		0010	0008		00570	89\$:	.WORD	90\$-89\$,-	:	
								91\$-89\$,-	:	
								93\$-89\$:	
			3E	11	00576		BRB	96\$:	1362
		00000000G	00	D4	00578	90\$:	CLRL	EDT\$\$G_DIR	:	1350
			21	11	0057E		BRB	94\$:	
			50	D4	00580	91\$:	CLRL	R0	:	1353
00000000G	00	00000000G	00	D1	00582		CMPL	EDT\$\$A_LN_PTR, EDT\$\$A_SEL_POS	:	
			02	1E	0058D		BGEQU	92\$:	

		50	D6	0058F		INCL	R0			
	00000000G	00	50	D0	00591	92\$:	MOVL	R0	EDT\$\$G_DIR	
			07	11	00598		BRB	94\$		
	00000000G	00	01	D0	0059A	93\$:	MOVL	#1	EDT\$\$G_DIR	1356
	00000000G	00	00	FB	005A1	94\$:	CALLS	#0	EDT\$\$SEL_RNGPOS	1373
02	FFFFFFFF	8F	50	CF	005A8		CASEL	R0, #1, #2		
0018		0021	000F		005B0	95\$:	.WORD	97\$-95\$,-		
								99\$-95\$,-		
								98\$-95\$		
	00000000G	00	00	FB	005B6	96\$:	CALLS	#0	EDT\$\$INTER_ERR	1389
			E2	11	005BD		BRB	94\$		1373
	00000000G	00	00	FB	005BF	97\$:	CALLS	#0	EDT\$\$CS_UP	1377
			D9	11	005C6		BRB	94\$		
	00000000G	00	00	FB	005C8	98\$:	CALLS	#0	EDT\$\$CS_DWN	1383
			D0	11	005CF		BRB	94\$		1373
	00000000G	00	00	D0	005D1	99\$:	MOVL	EDT\$\$A_SEL_POS, EDT\$\$A_LN_PTR		1397
	50	00000000G	00	9E	005DC		MOVAB	EDT\$\$T_LN_BUF, R0		1401
	50	00000000G	00	C2	005E3		SUBL2	EDT\$\$A_LN_PTR, R0		
	50		50	CE	005EA		MNEGL	R0, R0		
	000000FF	8F	50	D1	005ED		CMPL	R0, #255		
			07	1B	005F4		BLEQU	100\$		
	00000000G	00	00	FB	005F6		CALLS	#0	EDT\$\$INTER_ERR	
		00000000G	00	D4	005FD	100\$:	CLRL	EDT\$\$A_SEL_BUF		1407
			5B	D6	00603		INCL	SR		1408
			00CC	31	00605		BRW	118\$		1335
			52	D4	00608	101\$:	CLRL	CURSOR_MOVES		1417
		00000000G	00	D5	0060A		TSTL	EDT\$\$G_SEA_BEG		1419
			1D	12	00610		BNEQ	104\$		
	0B		58	D1	00612		CMPL	R8, #11		
			18	13	00615		BEQL	104\$		
	53	00000000G	00	D0	00617		MOVL	EDT\$\$G_SEA_STRLN, I		1422
			0C	11	0061E		BRB	103\$		
	00000000G	00	00	FB	00620	102\$:	CALLS	#0	EDT\$\$CS_LEFT	1424
	02		50	E9	00627		BLBC	R0, 103\$		
			52	D6	0062A		INCL	CURSOR_MOVES		
	F1		53	F4	0062C	103\$:	SOBGEQ	I, 102\$		
	01	08	AC	D1	0062F	104\$:	CMPL	COUNT, #1		1426
			4A	12	00633		BNEQ	109\$		
		00000000G	00	D5	00635		TSTL	EDT\$\$G_SEA_STRLN		1427
			42	15	0063B		BLEQ	109\$		
		00000000G	00	DD	0063D		PUSHL	EDT\$\$G_SEA_STRLN		
		00000000G	00	9F	00643		PUSHAB	EDT\$\$T_SEA_STR		
	00000000G	00	02	FB	00649		CALLS	#2	EDT\$\$TST_ONSTR	
	2C		50	E9	00650		BLBC	R0, 109\$		
		00000000G	00	D5	00653		TSTL	EDT\$\$G_SEA_BEG		1430
			1C	13	00659		BEQL	108\$		
	53	00000000G	00	D0	0065B		MOVL	EDT\$\$G_SEA_STRLN, I		1434
			07	11	00662		BRB	106\$		
	00000000G	00	00	FB	00664	105\$:	CALLS	#0	EDT\$\$CS_RIGHT	1435
	F6		53	F4	0066B	106\$:	SOBGEQ	I, 105\$		
	00000000G	00	01	D0	0066E	107\$:	MOVL	#1	EDT\$\$G_DIR	1437
			53	11	00675		BRB	116\$		1430
		00000000G	00	D4	00677	108\$:	CLRL	EDT\$\$G_DIR		1440
			4B	11	0067D		BRB	116\$		1430
		00000000G	00	D5	0067F	109\$:	TSTL	EDT\$\$G_SEA_BEG		1445
			11	12	00685		BNEQ	112\$		
	06		58	D1	00687		CMPL	R8, #6		

			0C	13	0068A		BEQL	112\$			
			07	11	0068C		BRB	111\$			1448
	00000000G	00	00	FB	0068E	110\$:	CALLS	#0, EDT\$\$CS_RIGHT			1449
		F6	52	F4	00695	111\$:	SOBGEQ	I, 110\$			
		03	58	D1	00698	112\$:	C MPL	R8, #3			1451
			0A	13	0069B		BEQL	113\$			
		04	58	D1	0069D		C MPL	R8, #4			
			05	13	006A0		BEQL	113\$			
		05	58	D1	006A2		C MPL	R8, #5			
			1A	12	006A5		BNEQ	115\$			
	00000000G	00	00	D5	006A7	113\$:	TSTL	EDT\$\$G_DIR			1454
			09	12	006AD		BNEQ	114\$			
	00000000G	00	00	FB	006AF		CALLS	#0, EDT\$\$CS_LEFT			
			12	11	006B6		BRB	116\$			
	00000000G	00	00	FB	006B8	114\$:	CALLS	#0, EDT\$\$CS_RIGHT			
			09	11	006BF		BRB	116\$			
			8F	DD	006C1	115\$:	PUSHL	#EDT\$NOSELN			1458
			039B	31	006C7		BRW	179\$			
		07	57	E9	006CA	116\$:	BLBC	SUCCEED, 118\$			1477
FAD3		01	08	AC	F1	006CD	117\$:	ACBL	COUNT, #1, I, 20\$		1018
			56	D5	006D4	118\$:	TSTL	NC			1487
			7C	13	006D6		BEQL	126\$			
		01	58	D1	006D8		C MPL	R8, #1			
			05	13	006DB		BEQL	119\$			
		02	58	D1	006DD		C MPL	R8, #2			
			72	12	006E0		BNEQ	126\$			
	50	59	02	C7	006E2	119\$:	DIVL3	#2, R9, R0			1490
	08	00	50	CF	006E6		CASEL	R0, #0, #8			
0040	0040	0040	001B		006EA	120\$:	.WORD	122\$-120\$,-			
0012	006C	006C	006C		006F2			124\$-120\$,-			
			006C		006FA			124\$-120\$,-			
								124\$-120\$,-			
								127\$-120\$,-			
								127\$-120\$,-			
								127\$-120\$,-			
								121\$-120\$,-			
								127\$-120\$			
	00000000G	00	00	FB	006FC	121\$:	CALLS	#0, EDT\$\$INTER_ERR			1512
			7D	11	00703		BRB	129\$			1490
			01	DD	00705	122\$:	PUSHL	#1			1495
			00000000G	00	9F	00707	PUSHAB	EDT\$\$T_DEL_CH			
			56	DD	0070D		PUSHL	NC			
	00000000G	00	03	FB	0070F		CALLS	#3, EDT\$\$SAV_DELTXT			1496
		50	56	D0	00716		MOVL	NC, R0			
		01	50	D1	00719		C MPL	R0, #1			
			03	15	0071C		BLEQ	123\$			
		50	01	D0	0071E		MOVL	#1, R0			
	00000000G	00	50	D0	00721	123\$:	MOVL	R0, EDT\$\$G_DEL_CHLEN			1490
			58	11	00728		BRB	129\$			1501
		7E	50	8F	9A	0072A	124\$:	MOVZBL	#80, -(SP)		
			00000000G	00	9F	0072E	PUSHAB	EDT\$\$T_DEL_WD			
			56	DD	00734		PUSHL	NC			
	00000000G	00	03	FB	00736		CALLS	#3, EDT\$\$SAV_DELTXT			1502
		50	56	D0	0073D		MOVL	NC, R0			
	00000050	8F	50	D1	00740		C MPL	R0, #80			
			04	15	00747		BLEQ	125\$			
		50	50	8F	9A	00749	MOVZBL	#80, R0			

		00000000G	00		50	D0	0074D	125\$:	MOVL	R0, EDT\$\$G_DEL_WDLN			
					2C	11	00754	126\$:	BRB	129\$		1490	
			7E	0100	8F	3C	00756	127\$:	MOVZWL	#256, -(SP)		1507	
				00000000G	00	9F	0075B		PUSHAB	EDT\$\$T_DEL_LN			
					56	DD	00761		PUSHL	NC			
		00000000G	00		03	FB	00763		CALLS	#3, EDT\$\$SAV_DELTXT		1508	
			50		56	D0	0076A		MOVL	NC, R0			
		00000100	8F		50	D1	0076D		CMPL	R0, #256			
					05	15	00774		BLEQ	128\$			
			50	0100	8F	3C	00776		MOVZWL	#256, R0			
		00000000G	00		50	D0	0077B	128\$:	MOVL	R0, EDT\$\$G_DEL_LNLEN			
			0F		59	D1	00782	129\$:	CMPL	R9, #15		1519	
					06	13	00785		BEQL	130\$			
				00000000G	00	D4	00787		CLRL	EDT\$\$G_VERT			
			51	0A	AE	B0	0078D	130\$:	MOVW	UPPER_WORD, SAVE		1524	
			50	00000000G	00	D0	00791		MOVL	EDT\$\$A_CUR_BUF, R0			
04	AE	0C	AE	06	A0	C3	00798		SUBL3	6(R0), ORIG_LNO, NUM_LINES			
		08	AE	10	AE	D0	0079F		MOVL	ORIG_LNO, NUM_LINES			
		08	AE	0A	A0	D9	007A4		SBWC	10(R0), NUM_LINES			
		0A	AE		51	B0	007A9		MOVW	SAVE, UPPER_WORD			
					08	AE	B5	007AD	TSTW	NUM_LINES+4		1526	
					19	18	007B0		BGEQ	131\$			
			51	0A	AE	B0	007B2		MOVW	UPPER_WORD, SAVE		1528	
04	AE	06	A0	0C	AE	C3	007B6		SUBL3	ORIG [NO, 6(R0), NUM_LINES			
		08	AE	0A	A0	D0	007BD		MOVL	10(R0), NUM_LINES			
		08	AE	10	AE	D9	007C2		SBWC	ORIG_LNO, NUM_LINES			
		0A	AE		51	B0	007C7		MOVW	SAVE, UPPER_WORD			
					04	AE	D6	007CB	131\$:	INCL	FIRST_LWORD	1530	
					03	12	007CE		BNEQ	132\$			
					08	AE	B6	007D0		INCW	NEXT_WORD		
			07		5A	E8	007D3	132\$:	BLBS	R10, 133\$		1537	
		00000000G	00		00	FB	007D6		CALLS	#0, EDT\$\$INTER_ERR			
			50	00000000G	00	9E	007DD	133\$:	MOVAB	EDT\$\$T_LN_BUF, R0		1538	
			50	00000000G	00	C2	007E4		SUBL2	EDT\$\$A_LN_PTR, R0			
			50		50	CE	007EB		MNEGL	R0, R0			
		000000FF	8F		50	D1	007EE		CMPL	R0, #255			
					07	1B	007F5		BLEQU	134\$			
		00000000G	00		00	FB	007F7		CALLS	#0, EDT\$\$INTER_ERR			
				00000000G	00	D5	007FE	134\$:	TSTL	EDT\$\$G_DIR		1541	
					2B	12	00804		BNEQ	137\$			
14	AE	24	AE		0E	28	00806		MOV3	#14, START_POS, END_POS		1544	
					50	D4	0080C		CLRL	R0		1548	
		00FF	8F	20	AE	B1	0080E		CMPL	END_POS+12, #255			
					02	1A	00814		BGTRU	135\$			
					50	D6	00816		INCL	R0			
			52		50	D2	00818	135\$:	MCOML	R0, R2			
			07		52	E9	0081B		BLBC	R2, 136\$			
		00000000G	00		00	FB	0081E		CALLS	#0, EDT\$\$INTER_ERR			
					24	AE	9F	00825	136\$:	PUSHAB	START_POS	1551	
		00000000G	00		01	FB	00828		CALLS	#1, EDT\$\$SAV_BUFPOS			
					2D	11	0082F		BRB	140\$		1541	
					14	AE	9F	00831	137\$:	PUSHAB	END_POS	1555	
		00000000G	00		01	FB	00834		CALLS	#1, EDT\$\$SAV_BUFPOS			
					50	D4	0083B		CLRL	R0		1559	
		00FF	8F	20	AE	B1	0083D		CMPL	END_POS+12, #255			
					02	1A	00843		BGTRU	138\$			
					50	D6	00845		INCL	R0			

			OC	12	00912	BNEQ	158\$			
		14	AE	9F	00914	PUSHAB	END_POS		1683	
00000000G	00		01	FB	00917	CALLS	#1, EDT\$\$RPOS			
			13	11	0091E	BRB	161\$			
	52	00000000G	00	D0	00920	158\$:	MOVL	EDT\$\$G_SEA_STRLN, I	1686	
			07	11	00927	BRB	160\$			
00000000G	00		00	FB	00929	159\$:	CALLS	#0, EDT\$\$CS_RIGHT	1687	
	F6		52	F4	00930	160\$:	SOBGEQ	I, 159\$		
		00000000G	00	D5	00933	161\$:	TSTL	EDT\$\$A_SEL_BUF	1689	
			28	13	00939	BEQL	164\$			
		00000000G	8F	DD	0093B	PUSHL	#EDT\$ SELALRACT		1692	
00000000G	00		01	FB	00941	CALLS	#1, EDT\$\$MSG_BELL			
	01	00000000G	00	D1	00948	CMPL	EDT\$\$G_DIR, #1		1694	
			05	12	0094F	BNEQ	162\$			
		24	AE	9F	00951	PUSHAB	START_POS			
			03	11	00954	BRB	163\$			
		14	AE	9F	00956	162\$:	PUSHAB	END_POS		
00000000G	00		01	FB	00959	163\$:	CALLS	#1, EDT\$\$RPOS		
			024D	31	00960	BRW	194\$		1696	
	56	00000000G	00	D0	00963	164\$:	MOVL	EDT\$\$A_CUR_BUF, R6	1700	
00000000G	00	06	A6	06	28	0096A	MOV3	#6, 6(R6), EDT\$\$L_SEL_LN		
			56	D0	00973	MOVL	R6, EDT\$\$A_SEL_BUF		1701	
00000000G	00	00000000G	00	D0	0097A	MOVL	EDT\$\$A_LN_PTR, EDT\$\$A_SEL_POS		1702	
			00	D5	00985	TSTL	EDT\$\$A_OLD_SEL		1708	
			07	13	0098B	BEQL	165\$			
00000000G	00		01	D0	0098D	MOVL	#1, EDT\$\$G_SCR_REBUILD			
			7E	D4	00994	165\$:	CLRL	-(SP)	1712	
		00000000G	00	DD	00996	PUSHL	EDT\$\$G_SEA_STRLN			
			01	DD	0099C	PUSHL	#1			
	F65D	CF	03	FB	0099E	CALLS	#3, EDT\$\$EXE_CHMCM2			
			64	11	009A3	BRB	171\$			
		08	AE	B5	009A5	166\$:	TSTW	NUM_LINES+4	1725	
			6D	12	009A8	BNEQ	174\$			
		06	AE	B5	009AA	TSTW	NUM_LINES+2			
			68	12	009AD	BNEQ	174\$			
			50	D4	009AF	CLRL	R0		1733	
		20	AE	B5	009B1	TSTW	END_POS+12			
			02	12	009B4	BNEQ	167\$			
			50	D6	009B6	INCL	R0			
		53	04	AE	3C	009B8	167\$:	MOVZWL	NUM_LINES, R3	
			53	50	C2	009BC	SUBL2	R0, R3		
				52	D4	009BF	CLRL	I		
			0E	11	009C1	BRB	169\$			
00000000G	00		00	FB	009C3	168\$:	CALLS	#0, EDT\$\$TADJ_CMD	1735	
00000000G	00		00	FB	009CA	CALLS	#0, EDT\$\$CS_DQN		1736	
EE	52		53	F3	009D1	169\$:	AOBLEQ	R3, I, 168\$	1733	
			35	11	009D5	BRB	172\$		1576	
		00000000G	00	D4	009D7	170\$:	CLRL	EDT\$\$A_ALT_BUF	1743	
			08	AE	B5	009DD	TSTW	NUM_LINES+4	1745	
			7D	12	009E0	BNEQ	178\$			
		06	AE	B5	009E2	TSTW	NUM_LINES+2			
			78	12	009E5	BNEQ	178\$			
		50	20	AE	3C	009E7	MOVZWL	END_POS+12, R0	1753	
			0040	9F	009EB	PUSHAB	EDT\$\$T_LN_BUF[R0]			
00000000G	00		7E	08	AE	3C	009F2	MOVZWL	NUM_LINES, -(SP)	
			00	02	FB	009F6	CALLS	#2, EDT\$\$DEL_TXTLN		
			02	58	D1	009FD	CMPL	R8, #2	1755	

			0A 12 00A00	BNEQ	172\$	
			00 FB 00A02	CALLS	#0, EDT\$\$INS_MOD	
			50 D0 00A09	MOVL	R0, SUCCEED	
			016D 31 00A0C	BRW	188\$	1576
		08	AE B5 00A0F	TSTW	NUM_LINES+4	1764
			4B 12 00A12	BNEQ	178\$	
		06	AE B5 00A14	TSTW	NUM_LINES+2	
			46 12 00A17	BNEQ	178\$	
			00 FB 00A19	CALLS	#0, EDT\$\$RPL_CHGDLN	1772
			50 D4 00A20	CLRL	R0	1773
		20	AE B5 00A22	TSTW	END_POS+12	
			02 12 00A25	BNEQ	175\$	
			50 D6 00A27	INCL	R0	
		51 04	AE 3C 00A29	MOVZWL	NUM_LINES, R1	
7E		51	50 C3 00A2D	SUBL3	R0, R1, -(SP)	
			01 01 FB 00A31	CALLS	#1, EDT\$\$FILL_TXT	
			013A 31 00A38	BRW	187\$	1774
			00 D0 00A3B	MOVL	EDT\$\$A_ALT_BUF, R0	1785
			28 13 00A42	BEQL	180\$	
			50 D1 00A44	CMPL	R0, EDT\$\$A_CUR_BUF	1787
			08 12 00A4B	BNEQ	177\$	
			00000000G 8F DD 00A4D	PUSHL	#EDT\$_ATT CUTAPP	1790
			10 11 00A53	BRB	179\$	
		08	AE B5 00A55	TSTW	NUM_LINES+4	1794
			05 12 00A58	BNEQ	178\$	
		06	AE B5 00A5A	TSTW	NUM_LINES+2	
			10 13 00A5D	BEQL	181\$	
			00000000G 8F DD 00A5F	PUSHL	#EDT\$_BADRANGE	1797
			01 01 FB 00A65	CALLS	#1, EDT\$\$MSG_BELL	
			0141 31 00A6C	BRW	194\$	1798
			00 D0 00A6F	MOVL	EDT\$\$A_CUR_BUF, SAVE_BUF	1802
			00 D0 00A76	MOVL	EDT\$\$A_ALT_BUF, EDT\$\$A_CUR_BUF	1803
			52 D4 00A81	CLRL	R2	1805
			0A 58 D1 00A83	CMPL	R8, #10	
			0B 12 00A86	BNEQ	182\$	
			52 D6 00A88	INCL	R2	
			00 FB 00A8A	CALLS	#0, EDT\$\$WF_BOT	
			07 11 00A91	BRB	183\$	
			00 FB 00A93	CALLS	#0, EDT\$\$WF_CLRBUF	
			00 FB 00A9A	CALLS	#0, EDT\$\$START_INS	1807
			59 D0 00AA1	MOVL	SAVE_BUF, EDT\$\$A_CUR_BUF	1808
			50 AE 3C 00AA8	MOVZWL	END_POS+12, R0	1809
		20	00000000G 0040 9F 00AAC	PUSHAB	EDT\$\$T_LN_BUF[R0]	
		7E	08 AE 3C 00AB3	MOVZWL	NUM_LINES, -(SP)	
			02 FB 00AB7	CALLS	#2, EDT\$\$DEL_TXTLN	
			00 FB 00ABE	CALLS	#0, EDT\$\$RPL_CHGDLN	1810
			50 00000000G 00 D0 00AC5	MOVL	EDT\$\$A_CUR_BUF, R0	1811
			51 00000000G 00 9E 00ACC	MOVAB	EDT\$\$T_LN_BUF, R1	
OC	AD		00000000G 00 51 A3 00AD3	SUBW3	R1, EDT\$\$A_LN_PTR, 12(R0)	
			00000000G 00 00 D0 00ADC	MOVL	EDT\$\$A_ALT_BUF, EDT\$\$A_CUR_BUF	1812
			00000000G 00 00 FB 00AE7	CALLS	#0, EDT\$\$RD_CURLN	1813
			00000000G 00 00 FB 00AEE	CALLS	#0, EDT\$\$END_INS	1814
			76 52 E9 00AF5	BLBC	R2, 186\$	1819
			52 04 AE 3C 00AFB	MOVZWL	NUM_LINES, I	1823
			07 11 00AFC	BRB	185\$	
			00 FB 00AFE	CALLS	#0, EDT\$\$RD_PV LN	1824
			F6 52 F4 00B05	SOBGEQ	I, 184\$	

	00000000G	00	00	FB	00B08	CALLS	#0, EDT\$\$GET_TXTLN	:	1826	
	00000000G	00	00	FB	00B0F	CALLS	#0, EDT\$\$RD_PVRLN	:	1828	
		55	50	E9	00B16	BLBC	R0, 186\$:		
		51	00000000G	00	D0	00B19	MOVL	EDT\$\$G_LN_LEN, R1	:	1831
		58	00000000G	00	D0	00B20	MOVL	EDT\$\$A_WK_LN, R8	:	
		56		68	9A	00B27	MOVZBL	(R8), R6	:	
50		51		56	C1	00B2A	ADDL3	R6, R1, R0	:	
	000000FF	8F		50	D1	00B2E	CMPL	R0, #255	:	
				37	14	00B35	BGTR	186\$:	
00000000G0046	00000000G	00	51	28	00B37	MOV3	R1, EDT\$\$T_LN_BUF, EDT\$\$T_LN_BUF[R6]	:	1835	
00000000G 00	07	A8	56	28	00B44	MOV3	R6, 7(R8), EDT\$\$T_LN_BUF	:	1837	
	00000000G	00	56	C0	00B4D	ADDL2	R6, EDT\$\$G_LN_LEN	:	1838	
	00000000G	00	00	FB	00B54	CALLS	#0, EDT\$\$DEL_CURLN	:	1839	
		00000000G	00	DD	00B5B	PUSHL	EDT\$\$G_LN_LEN	:	1840	
		00000000G	00	9F	00B61	PUSHAB	EDT\$\$T_LN_BUF	:		
	00000000G	00	02	FB	00B67	CALLS	#2, EDT\$\$RPL_LN	:		
	00000000G	00	59	D0	00B6E	186\$:	MOVL	SAVE_BUF, EDT\$\$A_CUR_BUF	:	1845
	00000000G	00	00	FB	00B75	187\$:	CALLS	#0, EDT\$\$GET_TXTLN	:	1846
			57	D5	00B7C	188\$:	TSTL	SUCCEED	:	1860
			2C	12	00B7E		BNEQ	193\$:	
		08	6E	E9	00B80		BLBC	(SP), 189\$:	1864
	00000000G		8F	DD	00B83		PUSHL	#EDT\$_STRNOTFND	:	
			1A	11	00B89		BRB	192\$:	
	00000000G		00	D5	00B8B	189\$:	TSTL	EDT\$\$G_DIR	:	1869
			09	12	00B91		BNEQ	190\$:	
	50 00000000G		8F	D0	00B93		MOVL	#EDT\$_TOPOFBUF, R0	:	
			07	11	00B9A		BRB	191\$:	
	50 00000000G		8F	D0	00B9C	190\$:	MOVL	#EDT\$_BOTOFBUF, R0	:	
			50	DD	00BA3	191\$:	PUSHL	R0	:	
	00000000G	00	01	FB	00BA5	192\$:	CALLS	#1, EDT\$\$MSG_BELL	:	1864
		50	57	D0	00BAC	193\$:	MOVL	SUCCEED, R0	:	1871
				04	00BAF		RET		:	
			50	D4	00BB0	194\$:	CLRL	R0	:	1872
				04	00BB2		RET		:	

: Routine Size: 2995 bytes, Routine Base: _EDT\$CODE + 0007

: 1305 1873 1

```

: 1307 1874 1 %SBTTL 'CHANGE_CASE - Change the case of characters in a range'
: 1308 1875 1 ROUTINE CHANGE_CASE (      | Change the case of characters in a range
: 1309 1876 1     NUM_LINES,                    | Number of lines to process
: 1310 1877 1     START_POS,                | Place to start in first line
: 1311 1878 1     END_POS                    | Place to stop in last line
: 1312 1879 1     ) : NOVALUE =
: 1313 1880 1
: 1314 1881 1 ++
: 1315 1882 1 FUNCTIONAL DESCRIPTION:
: 1316 1883 1
: 1317 1884 1     This routine scans over a range and changes the case of the characters
: 1318 1885 1     in that range.
: 1319 1886 1
: 1320 1887 1 FORMAL PARAMETERS:
: 1321 1888 1
: 1322 1889 1     NUM_LINES                the number of lines to process
: 1323 1890 1
: 1324 1891 1     START_POS                the character position in the first line at which we
: 1325 1892 1     should start
: 1326 1893 1
: 1327 1894 1     END_POS                the character position in the last line at which we should stop
: 1328 1895 1
: 1329 1896 1 IMPLICIT INPUTS:
: 1330 1897 1
: 1331 1898 1     EDT$$G_DIR
: 1332 1899 1     EDT$$T_LN_BUF
: 1333 1900 1     EDT$$A_LN_PTR
: 1334 1901 1     EDT$$A_LN_END
: 1335 1902 1
: 1336 1903 1 IMPLICIT OUTPUTS:
: 1337 1904 1
: 1338 1905 1     NONE
: 1339 1906 1
: 1340 1907 1 ROUTINE VALUE:
: 1341 1908 1
: 1342 1909 1     NONE
: 1343 1910 1
: 1344 1911 1 SIDE EFFECTS:
: 1345 1912 1
: 1346 1913 1     NONE
: 1347 1914 1
: 1348 1915 1 --
: 1349 1916 1
: 1350 1917 2 BEGIN
: 1351 1918 2
: 1352 1919 2 MAP
: 1353 1920 2     END_POS : REF POS_BLOCK;
: 1354 1921 2
: 1355 1922 2 EXTERNAL ROUTINE
: 1356 1923 2     EDT$$CS_DWN,                | Move down a line
: 1357 1924 2     EDT$$RPOS : NOVALUE,        | Restore the saved buffer position
: 1358 1925 2     EDT$$MRK_LNCHG : NOVALUE,    | Mark changes in a line
: 1359 1926 2     EDT$$CHG_CAS : NOVALUE;      | Change the case of characters
: 1360 1927 2
: 1361 1928 2 EXTERNAL
: 1362 1929 2     EDT$$G_DIR,                | The current direction.
: 1363 1930 2     EDT$$T_LN_BUF,            | Current line buffer

```

```

: 1364      1931      2      EDT$$A_LN_PTR,      ! Current character pointer
: 1365      1932      2      EDT$$A_LN_END;      ! End of current line pointer
: 1366      1933      2
: 1367      1934      2      L AL
: 1368      1935      2      LC;
: 1369      1936      2
: 1370      L 1937      2      %IF EXTRA_ASSERTS
: 1371      1938      2      %THEN
: 1372      1939      2      ASSERT (.END_POS [POS_CHAR_POS] LEQ 255);
: 1373      1940      2      %FI
: 1374      1941      2
: 1375      1942      2      !+
: 1376      1943      2      ! Loop through all lines.
: 1377      1944      2      !-
: 1378      1945      2
: 1379      1946      2      INCR I FROM 1 TO .NUM_LINES DO
: 1380      1947      2      BEGIN
: 1381      1948      2      !+
: 1382      1949      2      ! Set up pointer to last character in line.
: 1383      1950      2      !-
: 1384      1951      2      LC = CH$PTR (.EDT$$A_LN_END);
: 1385      1952      2
: 1386      1953      2      IF (.I EQL .NUM_LINES) THEN LC = CH$PTR (EDT$$T_LN_BUF, .END_POS [POS_CHAR_POS]);
: 1387      1954      2
: 1388      L 1955      2      %IF EXTRA_ASSERTS
: 1389      1956      2      %THEN
: 1390      1957      2      !+
: 1391      1958      2      ! We must not ask for a negative amount of text to be processed.
: 1392      1959      2      !-
: 1393      1960      2      ASSERT (.EDT$$A_LN_PTR LEQA .LC);
: 1394      1961      2      ASSERT ((.LC - EDT$$T_LN_BUF) LEQ 255);
: 1395      1962      2      %FI
: 1396      1963      2
: 1397      1964      2      EDT$$CHG_CAS (.EDT$$A_LN_PTR, .LC);
: 1398      1965      2      !+
: 1399      1966      2      ! Mark that part of the line as changed.
: 1400      1967      2      !-
: 1401      1968      2      EDT$$MRK_LNCHG (SCR_EDIT_MODIFY, .EDT$$A_LN_PTR - EDT$$T_LN_BUF);
: 1402      1969      2      EDT$$CS_DWN ();
: 1403      1970      2      END;
: 1404      1971      2
: 1405      1972      2      !+
: 1406      1973      2      ! If the direction was backward, then position to the start of the range.
: 1407      1974      2      ! If forward, position to the end of the range.
: 1408      1975      2      !-
: 1409      1976      2
: 1410      1977      2      IF (.EDT$$G_DIR EQL DIR_BACKWARD) THEN EDT$$RPOS (.START_POS) ELSE EDT$$RPOS (.END_POS);
: 1411      1978      2
: 1412      1979      1      END;      ! of routine CHANGE_CASE

```

.EXTRN EDT\$\$MRK_LNCHG, EDT\$\$CHG_CAS

01FC 0000 CHANGE_CASE:

58 0000000G 00 9E 00002

.WORD Save R2,R3,R4,R5,R6,R7,R8
MOVAB EDT\$\$INTER_ERR, R8

: 1875
:

	57	00000000G	00	9E	00009	MOVAB	EDT\$\$A_LN_PTR, R7	
	56	00000000G	00	9E	00010	MOVAB	EDT\$\$T_LN_BUF, R6	
	52	0C	AC	D0	00017	MOVL	END_POS, R2	1939
00FF	8F	0C	A2	B1	0001B	CMPW	12(R2), #255	
			03	1B	00021	BLEQU	1\$	
	68		00	FB	00023	CALLS	#0, EDT\$\$INTER_ERR	
	55	00FF	C6	9E	00026	MOVAB	EDT\$\$T_LN_BUF+255, R5	1961
			54	D4	0002B	CLRL	I	
			49	11	0002D	BRB	6\$	
	53	00000000G	00	D0	0002F	MOVL	EDT\$\$A_LN_END, LC	1951
04	AC		54	D1	00036	CMP	I, NUM_LINES	1953
			0A	12	0003A	BNEQ	3\$	
	50		66	9E	0003C	MOVAB	EDT\$\$T_LN_BUF, R0	
	53	0C	A2	3C	0003F	MOVZWL	12(R2), LC	
	53		50	C0	00043	ADDL2	R0, LC	
	53		67	D1	00046	CMP	EDT\$\$A_LN_PTR, LC	1960
			03	1B	00049	BLEQU	4\$	
	68		00	FB	0004B	CALLS	#0, EDT\$\$INTER_ERR	
	55		53	D1	0004E	CMP	LC, R5	1961
			03	15	00051	BLEQ	5\$	
	68		00	FB	00053	CALLS	#0, EDT\$\$INTER_ERR	
			53	DD	00056	PUSHL	LC	1964
			67	DD	00058	PUSHL	EDT\$\$A_LN_PTR	
00000000G	00		02	FB	0005A	CALLS	#2, EDT\$\$CHG_CAS	
	50		66	9E	00061	MOVAB	EDT\$\$T_LN_BUF, R0	1968
7E	67		50	C3	00064	SUBL3	R0, EDT\$\$A_LN_PTR, -(SP)	
			01	DD	00068	PUSHL	#1	
00000000G	00		02	FB	0006A	CALLS	#2, EDT\$\$MRK_LNCHG	
00000000G	00		00	FB	00071	CALLS	#0, EDT\$\$CS_DWN	1969
B2	54	04	AC	F3	00078	AOBLEQ	NUM_LINES, I, 2\$	1946
		00000000G	00	D5	0007D	TSTL	EDT\$\$G_DIR	1977
			05	12	00083	BNEQ	7\$	
			08	AC	DD	00085	PUSHL	START_POS
			02	11	00088	BRB	8\$	
00000000G	00		52	DD	0008A	PUSHL	R2	7\$:
			01	FB	0008C	CALLS	#1, EDT\$\$RPOS	8\$:
			04	00093		RET		1979

; Routine Size: 148 bytes, Routine Base: _EDT\$CODE + 0BBA

; 1413 1980 1 !<BLF/PAGE>

```

: 1415      1981  1 END
: 1416      1982  1
: 1417      1983  0 ELUDOM
! of module EDT$CHMEXVERB
    
```

PSECT SUMMARY

```

Name          Bytes          Attributes
_EDT$CODE     3150 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	97	25	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
_\$255\$DUA28:[EDT.SRC]SUPPORTS.L32;1	2	1	50	5	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:CHMEXVERB/OBJ=OBJ\$:CHMEXVERB MSRC\$:CHMEXVERB.BLI/UPDATE=(ENHS:C H MEXVERB)

```

: Size:          3143 code + 7 data bytes
: Run Time:      01:40.3
: Elapsed Time:  01:57.4
: Lines/CPU Min: 1186
: Lexemes/CPU-Min: 6435
: Memory Used:  604 pages
: Compilation Complete
    
```


CHMFINENT LIS	CHMINIT LIS				
CHMGOUNT LIS	CHMGINSTR LIS	CHMGSUSTR LIS	CHMINSMOD LIS		
CHMEMESS LIS	CHMINSTAB LIS				
CHMENTRM LIS	CHMINSCHR LIS				
CHMEXVERB LIS	CHMINSTR LIS	CHMGDTR LIS	CHMGQSTR LIS	CHMLPKPD LIS	CHMINSTR LIS
CHMENDWRD LIS	CHMKEYWRD LIS				
CHMEXCOM LIS					