


```

1 0001 0 %TITLE 'EDT$CHMPAREN - handle parentheses'
2 0002 0 MODULE EDT$CHMPAREN ( ! Handle parentheses
3 0003 0 IDENT = 'V04-000' ! File: CHMPAREN.BLI Edit: JBS1007
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 implements a parenthesized string of commands.
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: Unknown
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 04-Feb-1981. This module was created by
45 0045 1 extracting the routine EDT$SPARENT from module CHANGE.BLI.
46 0046 1 1-002 - Regularize headers. JBS 03-Mar-1981
47 0047 1 1-003 - Use new message codes. JBS 04-Aug-1981
48 0048 1 1-004 - Add return values. JBS 02-Oct-1981
49 0049 1 1-005 - Add an entry point so that this module can be loaded back into
50 0050 1 memory after calling the change mode parser. Also, use a different
51 0051 1 symbol for the name of the change mode parser so the call can be
52 0052 1 ignored by the overlay analyzer (and so that the call can be
53 0053 1 intercepted by a routine which will load us back into memory).
54 0054 1 JBS 02-Apr-1982
55 0055 1 1-006 - Set a flag if control C actually aborts something. JBS 24-May-1982
56 0056 1 1-007 - If an error occurs in processing the commands inside parens, such
57 0057 1 as string not found, continue after the right paren. JBS 23-Feb-1983

```



```
61 0060 1 %SBTTL 'Declarations'  
62 0061 1 :  
63 0062 1 : TABLE OF CONTENTS:  
64 0063 1 :  
65 0064 1 :  
66 0065 1 REQUIRE 'EDT$SRC:TRAROUNAM';  
67 0504 1 :  
68 0505 1 FORWARD ROUTINE  
69 0506 1 EDT$$PARENT,  
70 0507 1 EDT$$LOAD_CHMPAREN : NOVALUE;  
71 0508 1 :  
72 0509 1 :  
73 0510 1 : INCLUDE FILES:  
74 0511 1 :  
75 0512 1 :  
76 0513 1 REQUIRE 'EDT$SRC:EDTREQ';  
77 0648 1 :  
78 0649 1 :  
79 0650 1 : MACROS:  
80 0651 1 :  
81 0652 1 : NONE  
82 0653 1 :  
83 0654 1 : EQUATED SYMBOLS:  
84 0655 1 :  
85 0656 1 : NONE  
86 0657 1 :  
87 0658 1 : OWN STORAGE:  
88 0659 1 :  
89 0660 1 : NONE  
90 0661 1 :  
91 0662 1 : EXTERNAL REFERENCES:  
92 0663 1 :  
93 0664 1 : In the routine
```

```
! Collect and execute a parenthesized string of commands  
! Load this module into memory (on the PDP-11)
```

```

: 95 0665 1 %SBTTL 'EDT$$PARENT - handle parentheses'
: 96 0666 1
: 97 0667 1 GLOBAL ROUTINE EDT$$PARENT (           ! Handle parentheses
: 98 0668 1     COUNT                               ! Number of times to execute
: 99 0669 1     EXECUTE                             ! 1 = execute the commands
100 0670 1     ) =
101 0671 1
102 0672 1 !++
103 0673 1 ! FUNCTIONAL DESCRIPTION:
104 0674 1
105 0675 1     This routine implements a parenthesized string of commands. The
106 0676 1     string is analyzed without execution in order to verify it syntactically,
107 0677 1     and find the end of the string. It is then executed COUNT times, or until
108 0678 1     a command in the string fails. In either case, EDT$$A_CMD_BUF is left pointing
109 0679 1     at the first character after the parenthesized command list.
110 0680 1
111 0681 1 FORMAL PARAMETERS:
112 0682 1
113 0683 1     COUNT           Number of times to execute
114 0684 1
115 0685 1     EXECUTE         1 = execute the commands
116 0686 1
117 0687 1 IMPLICIT INPUTS:
118 0688 1
119 0689 1     EDT$$A_CMD_BUF
120 0690 1
121 0691 1 IMPLICIT OUTPUTS:
122 0692 1
123 0693 1     EDT$$G_CC_DONE
124 0694 1
125 0695 1 ROUTINE VALUE:
126 0696 1
127 0697 1     1 = ok, 0 = control C, 2 = end of journal file
128 0698 1
129 0699 1 SIDE EFFECTS:
130 0700 1
131 0701 1     NONE
132 0702 1
133 0703 1 --
134 0704 1
135 0705 2 BEGIN
136 0706 2
137 0707 2 EXTERNAL ROUTINE
138 0708 2     EDT$$MSG_BELL : NOVALUE,           ! Output a message to the terminal with a warning bell
139 0709 2     EDT$$CHK_CC,           ! Check to see if a CTRL/C has been typed
140 0710 2     EDT$$CHM_PAREXE_NOOVERLAY; ! Parse and execute a change mode command string
141 0711 2
142 0712 2 EXTERNAL
143 0713 2     EDT$$A_CMD_BUF,           ! Command string pointer
144 0714 2     EDT$$G_CC_DONE;         ! Set to 1 if control C actually aborts something
145 0715 2
146 0716 2 MESSAGES ((PARENMISS));
147 0717 2
148 0718 2 LOCAL
149 0719 2     SUCCEED,
150 0720 2     COM_END,
151 0721 2     COM_START;

```

```
152 0722 2
153 0723 2 EDT$$A_CMD_BUF = COM_START = CH$PLUS (.EDT$$A_CMD_BUF, 1);
154 0724 2
155 0725 2 + Check syntax and find the end of the command.
156 0726 2 -
157 0727 2 SUCCEED = EDT$$SCHM_PAREXE_NOOVERLAY (0);
158 0728 2
159 0729 2 IF (.SUCCEED NEQ 1) THEN RETURN (.SUCCEED);
160 0730 2
161 0731 2 COM_END = .EDT$$A_CMD_BUF;
162 0732 2 +
163 0733 2 - If we are not positioned on the right paren we have a syntax error.
164 0734 2
165 0735 2
166 0736 3 IF (CH$RCHAR (.COM_END) NEQ %C')')
167 0737 2 THEN
168 0738 3 BEGIN
169 0739 3 EDT$$MSG_BELL (EDT$_PARENMIS);
170 0740 3 RETURN (0);
171 0741 2 END;
172 0742 2
173 0743 2 IF .EXECUTE
174 0744 2 THEN
175 0745 2
176 0746 2 DECR I FROM .COUNT - 1 TO 0 DO
177 0747 3 BEGIN
178 0748 3
179 0749 3 IF EDT$$CHK_CC ()
180 0750 3 THEN
181 0751 4 BEGIN
182 0752 4 EDT$$G_CC_DONE = 1;
183 0753 4 RETURN (0);
184 0754 3 END;
185 0755 3
186 0756 3 EDT$$A_CMD_BUF = .COM_START;
187 0757 3 SUCCEED = EDT$$SCHM_PAREXE_NOOVERLAY (1);
188 0758 3
189 0759 3 IF (.SUCCEED NEQ 1) THEN EXITLOOP;
190 0760 3
191 0761 2 END;
192 0762 2
193 0763 2 EDT$$A_CMD_BUF = CH$PLUS (.COM_END, 1);
194 0764 2 +
195 0765 2 - Don't indicate an error in executing the command within parens as an error
196 0766 2 in executing this command. This allows multiple commands to be executed
197 0767 2 terminated by search failures.
198 0768 2 -
199 0769 2
200 0770 2 IF (.SUCCEED EQL 0) THEN SUCCEED = 1;
201 0771 2
202 0772 2 RETURN (.SUCCEED);
203 0773 1
```

! of routine EDT\$PARENT

.TITLE EDT\$CHMPAREN EDT\$CHMPAREN - handle parentheses
.IDENT \V04-000\

					.EXTRN EDT\$\$MSG_BELL, EDT\$\$CHK_CC	
					.EXTRN EDT\$\$SCHM_PAREXE_NOOVERLAY	
					.EXTRN EDT\$\$A_CMD_BUF, EDT\$\$G_CC_DONE	
					.EXTRN EDT\$_PARENMIS	
					.PSECT _EDT\$CODE, NOWRT, SHR, PIC, 2	
			00FC 00000		.ENTRY EDT\$\$PARENT, Save R2,R3,R4,R5,R6,R7	0667
57	00000000G	00	9E 00002		MOVAB EDT\$\$SCHM_PAREXE_NOOVERLAY, R7	
56	00000000G	00	9E 00009		MOVAB EDT\$\$A_CMD_BUF, R6	
66		01	C1 00010		ADDL3 #1, EDT\$\$A_CMD_BUF, COM_START	0723
66		54	D0 00014		MOVL COM_START, EDT\$\$A_CMD_BUF	
		7E	D4 00017		CLRL -(SP)	0727
67		01	FB 00019		CALLS #1, EDT\$\$SCHM_PAREXE_NOOVERLAY	
53		50	D0 0001C		MOVL R0, SUCCEED	
01		53	D1 0001F		CMPL SUCCEED, #1	0729
		52	12 00022		BNEQ 6\$	
55		66	D0 00024		MOVL EDT\$\$A_CMD_BUF, COM_END	0731
29		65	91 00027		CMPB (COM_END), #41	0736
		0F	13 0002A		BEQL 1\$	
	00000000G	00	00000000G 8F DD 0002C		PUSHL #EDT\$ PARENMIS	0739
		01	FB 00032		CALLS #1, EDT\$\$MSG_BELL	
		3F	11 00039		BRB 7\$	0740
		2C	08 AC E9 0003B	1\$:	BLBC EXECUTE, 5\$	0743
		52	04 AC D0 0003F		MOVL COUNT, I	0756
			23 11 00043		BRB 4\$	
	00000000G	00	00 FB 00045	2\$:	CALLS #0, EDT\$\$CHK_CC	0749
		09	50 E9 0004C		BLBC R0, 3\$	
	00000000G	00	01 D0 0004F		MOVL #1, EDT\$\$G_CC_DONE	0752
			22 11 00056		BRB 7\$	0753
		66	54 D0 00058	3\$:	MOVL COM_START, EDT\$\$A_CMD_BUF	0756
			01 DD 0005B		PUSHL #1	0757
		67	01 FB 0005D		CALLS #1, EDT\$\$SCHM_PAREXE_NOOVERLAY	
		53	50 D0 00060		MOVL R0, SUCCEED	
		01	53 D1 00063		CMPL SUCCEED, #1	0759
			03 12 00066		BNEQ 5\$	
	DA		52 F4 00068	4\$:	SOBGEQ I, 2\$	0746
	66	01	A5 9E 0006B	5\$:	MOVAB 1(R5), EDT\$\$A_CMD_BUF	0763
			53 D5 0006F		TSTL SUCCEED	0770
			03 12 00071		BNEQ 6\$	
		53	01 D0 00073		MOVL #1, SUCCEED	
		50	53 D0 00076	6\$:	MOVL SUCCEED, R0	0772
			04 00079		RET	
			50 D4 0007A	7\$:	CLRL R0	0773
			04 0007C		RET	

: Routine Size: 125 bytes, Routine Base: _EDT\$CODE + 0000

: 204 0774 1


```

: 206 0775 1 %SBTTL 'EDT$$LOAD_CHMPAREN - load this module into memory'
: 207 0776 1
: 208 0777 1 GLOBAL ROUTINE EDT$$LOAD_CHMPAREN ! Load this module into memory
: 209 0778 1 : NOVALUE =
: 210 0779 1
: 211 0780 1 ++
: 212 0781 1 FUNCTIONAL DESCRIPTION:
: 213 0782 1
: 214 0783 1 This routine has no function. It exists as an entry point so that
: 215 0784 1 EDT$$CHM_PAREXE_NOOVERLAY can call this module back into memory before
: 216 0785 1 returning to it.
: 217 0786 1
: 218 0787 1 FORMAL PARAMETERS:
: 219 0788 1
: 220 0789 1 NONE
: 221 0790 1
: 222 0791 1 IMPLICIT INPUTS:
: 223 0792 1
: 224 0793 1 NONE
: 225 0794 1
: 226 0795 1 IMPLICIT OUTPUTS:
: 227 0796 1
: 228 0797 1 NONE
: 229 0798 1
: 230 0799 1 ROUTINE VALUE:
: 231 0800 1
: 232 0801 1 NONE
: 233 0802 1
: 234 0803 1 SIDE EFFECTS:
: 235 0804 1
: 236 0805 1 NONE
: 237 0806 1
: 238 0807 1 --
: 239 0808 1
: 240 0809 2 BEGIN
: 241 0810 2 0
: 242 0811 1 END;

```

! of routine EDT\$\$LOAD_CHMPAREN

0000 0000
04 0002

.ENTRY EDT\$\$LOAD_CHMPAREN, Save nothing
RET

: 0777
: 0811

: Routine Size: 3 bytes, Routine Base: _EDT\$CODE + 007D

```

: 243 0812 1
: 244 0813 1 !<BLF/PAGE>

```

EDT\$CHMPAREN
V04-000

EDT\$CHMPAREN - handle parentheses
EDT\$\$LOAD_CHMPAREN - load this module into memo

K 5
16-Sep-1984 00:06:07
14-Sep-1984 12:22:40

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

Page 8
(5)

EDT
V04

: 246 0814 1 END
: 247 0815 1
: 248 0816 0 ELUDOM

! of module EDT\$CHMPAREN

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	128	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	1	0	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

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

: Size: 128 code + 0 data bytes
: Run Time: 00:12.7
: Elapsed Time: 00:16.3
: Lines/CPU Min: 3855
: Lexemes/CPU-Min: 10502
: Memory Used: 79 pages
: Compilation Complete

