

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT

```

FFFFFFFFF  CCCCCCCC  HH      HH      AAAAAA  RRRRRRRR
FFFFFFFFF  CCCCCCCC  HH      HH      AAAAAA  RRRRRRRR
FF         CC        HH      HH      AA       AA  RR       RR
FF         CC        HH      HH      AA       AA  RR       RR
FF         CC        HH      HH      AA       AA  RR       RR
FF         CC        HH      HH      AA       AA  RR       RR
FFFFFFFFF  CC        HH      HH      AA       AA  RRRRRRRR
FFFFFFFFF  CC        HH      HH      AA       AA  RRRRRRRR
FF         CC        HH      HH      AAAAAAAAAA  RR  RR
FF         CC        HH      HH      AAAAAAAAAA  RR  RR
FF         CC        HH      HH      AA       AA  RR       RR
FF         CC        HH      HH      AA       AA  RR       RR
FF         CC        HH      HH      AA       AA  RR       RR
FF         CC        HH      HH      AA       AA  RR       RR

```

```

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

```

```

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

```



```
0001 0 %TITLE 'EDT$FCHAR - put a char in format buffer'
0002 0 MODULE EDT$FCHAR ( ! Put a char in format buffer
0003 0 IDENT = 'V04-000' ! File: FCHAR.BLI Edit: JBS1015
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 Put a character in the option buffer, expanding control characters
0037 1 and watching for line overflow.
0038 1
0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0040 1
0041 1 AUTHOR: Bob Kushlis, CREATION DATE: March 18, 1979
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 1-001 - Original. DJS 19-FEB-1981. This module was created by
0046 1 extracting routine EDT$$FMT.CH from module FORMAT.
0047 1 1-002 - Regularize headers. JBS 05-Mar-1981
0048 1 1-003 - Change output of <FF> to user string. STS 07-Oct-1981
0049 1 1-004 - Don't count <FF>'s width twice. JBS 05-May-1982
0050 1 1-005 - Correct appearance of <CR>. JBS 07-May-1982
0051 1 1-006 - Add supplemental set from DEC STD 169. JBS 11-Aug-1982
0052 1 1-007 - Update EDT$$G.PRIV.COL. JBS 30-Sep-1982
0053 1 1-008 - Remove external declaration of EDT$$FMT.LIT, not used. JBS 05-Oct-1982
0054 1 1-009 - Don't increment EDT$$G.PRIV.COL beyond the size of the screen. JBS 16-Oct-1982
0055 1 1-010 - Don't output the buffer based on the terminal's width. JBS 16-Oct-1982
0056 1 1-011 - Remove optimization of simple characters, now done by caller. JBS 04-Jan-1983
0057 1 1-012 - Add conditional for VT220 support. JBS 10-Feb-1983
```

EDTSFCHAR
V04-000

EDTSFCHAR - put a char in format buffer

6 9
16-Sep-1984 00:19:02
14-Sep-1984 12:23:04

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

Page 2
(1)

:	58	0058	1	:	1-013	-	Take out unnecessary declarations.	SMB	23-Feb-1983
:	59	0059	1	:	1-014	-	Put character names in DATA and revise the format of the table.	JBS	04-Mar-1983
:	60	0060	1	:	1-015	-	Correct display on 8-bit terminals.	JBS	07-Mar-1983
:	61	0061	1	:	--				
:	62	0062	1	:					

EDT\$FCHAR
V04-000

EDT\$FCHAR - put a char in format buffer
Declarations

D 9
16-Sep-1984 00:19:02
14-Sep-1984 12:23:04

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

```
: 64      0063 1 %SBTTL 'Declarations'
: 65      0064 1
: 66      0065 1 | TABLE OF CONTENTS:
: 67      0066 1 |
: 68      0067 1
: 69      0068 1 REQUIRE 'EDT$SRC:TRAROUNAM';
: 70      0507 1
: 71      0508 1 FORWARD ROUTINE
: 72      0509 1     EDT$FMT_CH : NOVALUE;
: 73      0510 1
: 74      0511 1 |
: 75      0512 1 | INCLUDE FILES:
: 76      0513 1 |
: 77      0514 1
: 78      0515 1 REQUIRE 'EDT$SRC:EDTREQ';
: 79      0650 1
: 80      0651 1 LIBRARY 'EDT$SRC:SUPPORTS';
: 81      0652 1
: 82      0653 1 LIBRARY 'EDT$SRC:TRANSLATE';
: 83      0654 1
: 84      0655 1 |
: 85      0656 1 | MACROS:
: 86      0657 1 |
: 87      0658 1 |     NONE
: 88      0659 1 |
: 89      0660 1 | EQUATED SYMBOLS:
: 90      0661 1 |
: 91      0662 1 |     NONE
: 92      0663 1 |
: 93      0664 1 | OWN STORAGE:
: 94      0665 1 |
: 95      0666 1 |     NONE
: 96      0667 1 |
: 97      0668 1 | EXTERNAL REFERENCES:
: 98      0669 1 |
: 99      0670 1 |     In the routine
```

```

: 101 0671 1 %SBTTL 'EDT$$FMT_CH - put a char in format buffer'
: 102 0672 1
: 103 0673 1 GLOBAL ROUTINE EDT$$FMT_CH (          ! Put a char in the format buffer
: 104 0674 1     FC                                ! Character to print
: 105 0675 1     ) : NOVALUE =
: 106 0676 1
: 107 0677 1 !++
: 108 0678 1 ! FUNCTIONAL DESCRIPTION:
: 109 0679 1
: 110 0680 1 !     Place a character in the format buffer.  If the character would cause
: 111 0681 1 !     the buffer to overflow, or a line to be longer than the terminal width,
: 112 0682 1 !     then write the buffer first.  Control characters are printed out either
: 113 0683 1 !     with a special mnemonic like <CR> or as ^letter.  Tabs are expanded into
: 114 0684 1 !     the correct number of spaces.  If this is not an eight-bit terminal,
: 115 0685 1 !     all characters above 127 are printed using a name in <>.  If this is an
: 116 0686 1 !     eight-bit terminal controls above 127 are printed as <mnemonic>, reserved
: 117 0687 1 !     positions above 127 are printed as <Xnn>, where nn is the hex for the
: 118 0688 1 !     character.
: 119 0689 1
: 120 0690 1 ! FORMAL PARAMETERS:
: 121 0691 1
: 122 0692 1 !     FC                                The character to print
: 123 0693 1
: 124 0694 1 ! IMPLICIT INPUTS:
: 125 0695 1
: 126 0696 1 !     EDT$$G_TI_WID
: 127 0697 1 !     EDT$$G_EIGHT_BIT
: 128 0698 1 !     EDT$$G_FMT_LNPOS
: 129 0699 1
: 130 0700 1 ! IMPLICIT OUTPUTS:
: 131 0701 1
: 132 0702 1 !     EDT$$G_FMT_LNPOS
: 133 0703 1 !     EDT$$G_PRV_COL
: 134 0704 1
: 135 0705 1 ! ROUTINE VALUE:
: 136 0706 1
: 137 0707 1 !     NONE
: 138 0708 1
: 139 0709 1 ! SIDE EFFECTS:
: 140 0710 1
: 141 0711 1 !     NONE
: 142 0712 1
: 143 0713 1 ! --
: 144 0714 1
: 145 0715 2 ! BEGIN
: 146 0716 2
: 147 0717 2 ! EXTERNAL ROUTINE
: 148 0718 2 !     EDT$$FMT_TEXT : NOVALUE,          ! Output the text for form feed
: 149 0719 2 !     EDT$$STORE_FMTCH : NOVALUE;       ! Put a character in the format buffer
: 150 0720 2
: 151 0721 2 ! EXTERNAL
: 152 0722 2 !     EDT$$G_TI_WID,                    ! Terminal width
: 153 0723 2 !     EDT$$G_EIGHT_BIT,                ! Is terminal in eight-bit mode?
: 154 0724 2 !     EDT$$G_FMT_LNPOS,                ! The current column number
: 155 0725 2
: 156 L 0726 2 ! IF SUPPORT_VT220
: 157 0727 2 ! THEN
```



```

158      0728 2      EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE], ! Information about each character
159      U 0729 2      %ELSE
160      U 0730 2      EDT$$B_CHAR_INFO : BLOCKVECTOR [128, 1, BYTE], ! Information about each character
161      0731 2      %FI
162      0732 2
163      0733 2      EDT$$A_CHAR_NAMES, ! Names of some characters
164      0734 2      EDT$$K_CHAR_NAMES_LEN, ! Length of the name table
165      0735 2      EDT$$G_PRV_COL; ! Cursor column
166      0736 2
167      0737 2      LOCAL
168      0738 2      C;
169      0739 2
170      0740 2      C = .FC;
171      0741 2
172      L 0742 3      %IF ( NOT SUPPORT_VT220)
173      U 0743 2      %THEN
174      U 0744 2      C = .C AND %X'7F';
175      0745 2      %FI
176      0746 2
177      0747 2      !+
178      0748 2      !- Watch for special cases.
179      0749 2      !-
180      0750 2
181      0751 2      SELECTONE .C OF
182      0752 2      SET
183      0753 2
184      0754 2      [ASC_K_TAB] :
185      0755 2      BEGIN
186      0756 2
187      0757 3      DO
188      0758 4      BEGIN
189      0759 4      EDT$$FMT_CH (%C' ');
190      0760 4      END
191      0761 4      UNTIL ((.EDT$$G_FMT_LNPOS AND 7) EQL 0)
192      0762 4
193      0763 2      END;
194      0764 2
195      0765 2      [ASC_K_FF] :
196      0766 2      BEGIN
197      0767 2      !+
198      0768 2      !- Handle form feed specially.
199      0769 2      !-
200      0770 2      EDT$$FMT_TEXT (1);
201      0771 2      END;
202      0772 2
203      0773 2      [OTHERWISE] :
204      0774 2      BEGIN
205      0775 2      !+
206      0776 2      !- This is not a special case character, dispatch on its type.
207      0777 2      !-
208      0778 2
209      0779 2      CASE .EDT$$B_CHAR_INFO [.C, 0, 2, 2, 0] FROM 0 TO 3 OF
210      0780 2      SET
211      0781 2
212      0782 2      [0] :
213      0783 2      BEGIN
214      0784 2      !+
```

EDT\$FCHAR
V04-000

EDT\$FCHAR - put a char in format buffer
EDT\$\$FMT_CH - put a char in format buffer

G 9
16-Sep-1984 00:19:02
14-Sep-1984 12:23:04

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

Page 6
(3)

```

: 215 0785 4 ! This is a simple character; it can be printed on this terminal in one column.
: 216 0786 4 ! Bump the column number by the amount occupied by this character.
: 217 0787 4 !-
: 218 0788 4 EDT$$G_FMT_LNPOS = .EDT$$G_FMT_LNPOS + 1;
: 219 0789 4 EDT$$STORE_FMTCH (.C);
: 220 0790 4
: 221 0791 4 IF (.EDT$$G_PRV_COL NEQ (.EDT$$G_TI_WID - 1)) THEN EDT$$G_PRV_COL = .EDT$$G_PRV_COL + 1;
: 222 0792 4
: 223 0793 3 END;
: 224 0794 3
: 225 0795 3 [1] :
: 226 0796 4 BEGIN
: 227 0797 4 !+
: 228 0798 4 ! This character is to be output as ^ followed by the character code plus 64.
: 229 0799 4 !-
: 230 0800 4 EDT$$FMT_CH (%C'^');
: 231 0801 4 EDT$$FMT_CH (.C + 64);
: 232 0802 4 END;
: 233 0803 3
: 234 0804 3 [2] :
: 235 0805 4 BEGIN
: 236 0806 4 !+
: 237 0807 4 ! This character has a special text form. Find it in the table
: 238 0808 4 ! and output the special form surrounded by <>. However, characters above the C1 controls are
: 239 0809 4 ! output as themselves on 8-bit terminals.
: 240 0810 4 !-
: 241 0811 4
: 242 0812 4 LOCAL
: 243 0813 4 REP_PTR,
: 244 0814 4 REP_CHAR;
: 245 0815 4
: 246 L 0816 4 %IF SUPPORT_VT220
: 247 0817 4 %THEN
: 248 0818 4
: 249 0819 5 IF (.EDT$$G_EIGHT_BIT AND (.C GEQ %X'AO'))
: 250 0820 4 THEN
: 251 0821 5 BEGIN
: 252 0822 5 !+
: 253 0823 5 ! This is a legitimate character in the DEC Multinational supplemental set, being displayed
: 254 0824 5 ! on an eight-bit terminal.
: 255 0825 5 !-
: 256 0826 5 EDT$$G_FMT_LNPOS = .EDT$$G_FMT_LNPOS + 1;
: 257 0827 5 EDT$$STORE_FMTCH (.C);
: 258 0828 5
: 259 0829 6 IF (.EDT$$G_PRV_COL NEQ (.EDT$$G_TI_WID - 1))
: 260 0830 5 THEN
: 261 0831 5 EDT$$G_PRV_COL = .EDT$$G_PRV_COL + 1;
: 262 0832 5
: 263 0833 5 END
: 264 0834 4 ELSE
: 265 0835 4 %FI
: 266 0836 4
: 267 0837 5 BEGIN
: 268 0838 5 REP_PTR = CH$PLUS (CH$FIND_CH (EDT$$K_CHAR_NAMES_LEN, EDT$$A_CHAR_NAMES, .C), 1);
: 269 0839 5 EDT$$FMT_CH (%C'<');
: 270 0840 5 REP_CHAR = CH$RCHAR_A (REP_PTR);
: 271 0841 5
```


EDT\$FCHAR
V04-000

EDT\$FCHAR - put a char in format buffer
EDT\$\$FMT_CH - put a char in format buffer

H 9
16-Sep-1984 00:19:02
14-Sep-1984 12:23:04

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

Page 7
(3)

```

272 0842 5      WHILE ((.REP_CHAR GEQ %X'20') AND (.REP_CHAR LEQ %X'7E')) DO
273 0843 6      BEGIN
274 0844 6      EDT$$FMT_CH (.REP_CHAR);
275 0845 6      REP_CHAR = CH$RCHAR_A (REP_PTR);
276 0846 5      END;
277 0847 5
278 0848 5      EDT$$FMT_CH (%C'>');
279 0849 4      END;
280 0850 4
281 0851 3      END;
282 0852 3
283 0853 3      [3] : BEGIN
284 0854 4
285 0855 4      !+ This character is to be output as <Xnn>, where nn is the hex for the character.
286 0856 4      !-
287 0857 4
288 0858 4      LOCAL
289 0859 4      HEX_DIGIT_1,
290 0860 4      HEX_DIGIT_2;
291 0861 4
292 0862 4      EDT$$FMT_CH (%C'<');
293 0863 4      EDT$$FMT_CH (%C'X');
294 0864 4      HEX_DIGIT_1 = (.C^4) + %C'0';
295 0865 4
296 0866 4      IF (.HEX_DIGIT_1 GTR %C'9') THEN HEX_DIGIT_1 = .HEX_DIGIT_1 - %C'9' + %C'A' - 1;
297 0867 4
298 0868 4      EDT$$FMT_CH (.HEX_DIGIT_1);
299 0869 4      HEX_DIGIT_2 = (.C^4 AND %X'0F') + %C'0';
300 0870 4
301 0871 4      IF (.HEX_DIGIT_2 GTR %C'9') THEN HEX_DIGIT_2 = .HEX_DIGIT_2 - %C'9' + %C'A' - 1;
302 0872 4
303 0873 4      EDT$$FMT_CH (.HEX_DIGIT_2);
304 0874 4      EDT$$FMT_CH (%C'>');
305 0875 4      END;
306 0876 3      TES;
307 0877 3
308 0878 3      END;
309 0879 2      TES;
310 0880 2
311 0881 2      END;
312 0882 1

```

! of routine EDT\$\$FMT_CH

```

.TITLE EDT$FCHAR EDT$FCHAR - put a char in format buff
.IDENT \V04-000\
.EXTRN EDT$$FMT_TEXT, EDT$$STORE_FMTCH
.EXTRN EDT$$G_TT_WID, EDT$$G_EIGHT_BIT
.EXTRN EDT$$G_FMT_LNPOS
.EXTRN EDT$$B_CHAR_INFO
.EXTRN EDT$$A_CHAR_NAMES
.EXTRN EDT$$K_CHAR_NAMES_LEN
.EXTRN EDT$$G_PRV_COL
.PSECT _EDT$CODE,NOWRT, SHR, PIC,2

```

EDT\$FCHAR
V04-000

EDT\$FCHAR - put a char in format buffer
EDT\$\$FMT_CH - put a char in format buffer

I 9
16-Sep-1984 00:19:02
14-Sep-1984 12:23:04

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

Page 8
(3)

			01FC 00000	.ENTRY	EDT\$\$FMT_CH, Save R2,R3,R4,R5,R6,R7,R8	0673
58	00000000G	00	9E 00002	MOVAB	EDT\$\$G_PRV_COL, R8	
57	00000000G	00	9E 00009	MOVAB	EDT\$\$G_FMT_LNPOS, R7	
56	ED	AF	9E 00010	MOVAB	EDT\$\$FMT_CH, R6	
52	04	AC	D0 00014	MOVL	FC, C	0740
09		52	D1 00018	CMPL	C, #9	0754
		0B	12 0001B	BNEQ	2\$	
		20	DD 0001D	PUSHL	#32	0759
66		01	FB 0001F	CALLS	#1, EDT\$\$FMT_CH	
07		67	93 00022	BITB	EDT\$\$G_FMT_LNPOS, #7	0761
		F6	12 00025	BNEQ	1\$	
			04 00027	RET		0755
0C		52	D1 00028	CMPL	C, #12	0765
		0A	12 0002B	BNEQ	3\$	
		01	DD 0002D	PUSHL	#1	0770
	00000000G	00	01 FB 0002F	CALLS	#1, EDT\$\$FMT_TEXT	
			04 00036	RET		0751
55	00000000G0042	02	02 EF 00037	EXTZV	#2, #2, EDT\$\$B_CHAR_INFO[C], R5	0779
	03	00	55 CF 00041	CASEL	R5, #0, #3	
0073	0017	000A	0027 00045	.WORD	7\$-4\$,- 5\$-4\$,- 6\$-4\$,- 11\$-4\$	
					7\$	0788
		1D 11 0004D	BRB			0800
7E	5E	8F 9A 0004F	MOVZBL	#94, -(SP)		
66		01 FB 00053	CALLS	#1, EDT\$\$FMT_CH		
	40	A2 9F 00056	PUSHAB	64(C)		0801
		0094 31 00059	BRW	15\$		
	000000A0	24 00000000G	00 E9 0005C	BLBC	EDT\$\$G_EIGHT_BIT, 8\$	0819
		8F	52 D1 00063	CMPL	C, #160	
			1B 19 0006A	BLSS	8\$	
			67 D6 0006C	INCL	EDT\$\$G_FMT_LNPOS	0826
			52 DD 0006E	PUSHL	C	0827
	00000000G	00	01 FB 00070	CALLS	#1, EDT\$\$STORE_FMTCH	
50	00000000G	00	01 C3 00077	SUBL3	#1, EDT\$\$G_TI_WID, R0	0829
		50	68 D1 0007F	CMPL	EDT\$\$G_PRV_COL, R0	
			6F 13 00082	BEQL	16\$	
			68 D6 00084	INCL	EDT\$\$G_PRV_COL	0831
			04 00086	RET		0819
	00000000G	50 00000000G	00 9E 00087	MOVAB	EDT\$\$K_CHAR_NAMES_LEN, R0	0838
		50	52 3A 0008E	LOCC	C, R0, EDT\$\$A_CHAR_NAMES	
			02 12 00096	BNEQ	9\$	
			51 D4 00098	CLRL	R1	
		54	A1 9E 0009A	MOVAB	1(R1), REP_PTR	0839
			3C DD 0009E	PUSHL	#60	
		66	01 FB 000A0	CALLS	#1, EDT\$\$FMT_CH	
		53	84 9A 000A3	MOVZBL	(REP_PTR)+, REP_CHAR	0840
		20	53 D1 000A6	CMPL	REP_CHAR, #32	0842
			43 19 000A9	BLSS	14\$	
	0000007E	8F	53 D1 000AB	CMPL	REP_CHAR, #126	
			3A 14 000B2	BGTR	14\$	
			53 DD 000B4	PUSHL	REP_CHAR	0844
			E8 11 000B6	BRB	10\$	
			3C DD 000B8	PUSHL	#60	0863
		66	01 FB 000BA	CALLS	#1, EDT\$\$FMT_CH	
		7E	8F 9A 000BD	MOVZBL	#88, -(SP)	0864
		66	01 FB 000C1	CALLS	#1, EDT\$\$FMT_CH	

EDT\$FCHAR
V04-000

EDT\$FCHAR - put a char in format buffer
EDT\$\$FMT_CH - put a char in format buffer

J 9
16-Sep-1984 00:19:02
14-Sep-1984 12:23:04

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

Page 9
(3)

EDT
V04

50	52	FC	8F	78	000C4	ASHL	#-4, C, R0	: 0865
	50		30	C0	000C9	ADDL2	#48, HEX_DIGIT_1	: 0867
	39		50	D1	000CC	CMPL	HEX_DIGIT_1, #57	: 0869
			03	15	000CF	BLEQ	12\$	: 0870
	50		07	C0	000D1	ADDL2	#7, HEX_DIGIT_1	: 0872
			50	DD	000D4	PUSHL	HEX_DIGIT_1	: 0874
50	52		01	FB	000D6	CALLS	#1, EDT\$\$FMT_CH	: 0875
	04		00	EF	000D9	EXTZV	#0, #4, C, HEX_DIGIT_2	: 0882
	50		30	C0	000DE	ADDL2	#48, HEX_DIGIT_2	: 0883
	39		50	D1	000E1	CMPL	HEX_DIGIT_2, #57	: 0884
			03	15	000E4	BLEQ	13\$	: 0885
	50		07	C0	000E6	ADDL2	#7, HEX_DIGIT_2	: 0886
			50	DD	000E9	PUSHL	HEX_DIGIT_2	: 0887
	66		01	FB	000EB	CALLS	#1, EDT\$\$FMT_CH	: 0888
			3E	DD	000EE	PUSHL	#62	: 0889
	66		01	FB	000F0	CALLS	#1, EDT\$\$FMT_CH	: 0890
			04	000F3	16\$:	RET		: 0891

; Routine Size: 244 bytes, Routine Base: _EDT\$CODE + 0000

: 313 0883 1
: 314 0884 1 !<BLF/PAGE>

EDT\$FCHAR
V04-000

EDT\$FCHAR - put a char in format buffer
EDT\$FMT_CH - put a char in format buffer

K 9
16-Sep-1984 00:19:02
14-Sep-1984 12:23:04

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

Page 10
(4)

: 316 0885 1 END
: 317 0886 1
: 318 0887 0 ELUDOM

! of module EDT\$FCHAR

PSECT SUMMARY

Name Bytes Attributes
_EDT\$CODE 244 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	2	0	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
-\$255\$DUA28:[EDT.SRC]TRANSLATE.L32;1	6	0	0	57	00:00.2

COMMAND QUALIFIERS

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

: Size: 244 code + 0 data bytes
: Run Time: 00:16.9
: Elapsed Time: 00:20.2
: Lines/CPU Min: 3150
: Lexemes/CPU-Min: 8994
: Memory Used: 109 pages
: Compilation Complete

0133 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXTEND
LIS

FDEC
LIS

FILL
LIS

FINDPARA
LIS

FCRLF
LIS

EDT
LIS

EXEC
LIS

EXECNOO
LIS

FILEIO
LIS

EDTVECTOR
LIS

FINDKEY
LIS

FCOLINC
LIS

FINAL
LIS

FINDHDLR
LIS

DEFKEY
LIS

ERRMSG
LIS

FCHAR
LIS