



\*\*FILE\*\*ID\*\*FCIMRA

F 4

FFFFFFFFF	CCCCCCC	IIIII	MM	MM	RRRRRRR	AAAAAA
FFFFFFFFF	CCCCCCC	IIII	MM	MM	RRRRRRR	AAAAAA
FF	CC	II	MMMM	MMMM	RR	RR AA
FF	CC	II	MMMM	MMMM	RR	RR AA
FF	CC	II	MM	MM	RR	RR AA
FF	CC	II	MM	MM	RR	RR AA
FFFFFFF	CC	II	MM	MM	RRRRRRR	AA
FFFFFFF	CC	II	MM	MM	RRRRRRR	AA
FF	CC	II	MM	MM	RR RR	AAAAAAA
FF	CC	II	MM	MM	RR RR	AAAAAAA
FF	CC	II	MM	MM	RR RR	RR AA
FF	CC	II	MM	MM	RR RR	AA AA
FF	CCCCCCC	IIIII	MM	MM	RR RR	AA AA
FF	CCCCCCC	IIIII	MM	MM	RR RR	AA AA

....

LL	IIIII	SSSSSSS
LL	IIII	SSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLLL	IIIII	SSSSSSS
LLLLLLLLL	IIIII	SSSSSSS

FIG  
V04

```
: 1      0001 0 MODULE fcimra ( IDENT = 'V04-000'  
1      0002 0           %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,  
1      0003 0           NONEXTERNAL = LONG_RELATIVE)]  
1      0004 0           ) =  
1      0005 1 BEGIN  
1      0006 1 *****  
1      0007 1 *  
1      0008 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
1      0009 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
1      0010 1 *  ALL RIGHTS RESERVED.  
1      0011 1 *  
1      0012 1 *  
1      0013 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
1      0014 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
1      0015 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
1      0016 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
1      0017 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
1      0018 1 *  TRANSFERRED.  
1      0019 1 *  
1      0020 1 *  
1      0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
1      0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
1      0023 1 *  CORPORATION.  
1      0024 1 *  
1      0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
1      0026 1 *  
1      0027 1 *  
1      0028 1 *****  
1      0029 1 :  
1      0030 1 :  
1      0031 1 ++  
1      0032 1 : FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS  
1      0033 1 :  
1      0034 1 : ABSTRACT:  
1      0035 1 :  
1      0036 1 :     Special handling for first character written into the MRA.  
1      0037 1 :  
1      0038 1 : ENVIRONMENT: Transportable  
1      0039 1 :  
1      0040 1 : AUTHOR: R.W.Friday CREATION DATE: June, 1978  
1      0041 1 :
```

## Revision History

H 4  
16-Sep-1984 00:27:05    VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 13:06:11    [RUNOFF.SRC]FCIMRA.BLI;1

Page 2  
(2)FIG  
V04

43 0042 1 %SBTTL 'Revision History'  
44 0043 1 MODIFIED BY:  
45 0044 1  
46 0045 1 011 KFA00011 Ken Alden 05-Oct-1983  
47 0046 1 For DSRPLUS: Added bit sca margin pad to help the CREF  
48 0047 1 facility determine what is in the MRA.  
49 0048 1  
50 0049 1 010 RER00010 Ron Randall 12-May-1983  
51 0050 1 For DSRPLUS: Fixed numbered-footnote/entity-resolution bug.  
52 0051 1  
53 0052 1 009 RER00009 Ron Randall 18-Apr-1983  
54 0053 1 Fixed left margin indentation bug.  
55 0054 1  
56 0055 1 008 RER00008 Ron Randall 06-Apr-1983  
57 0056 1 For DSRPLUS:  
58 0057 1 Adds footnote number or blanks at the start of each footnote  
59 0058 1 line being collected if footnote numbering is on.  
60 0059 1  
61 0060 1 007 RER00007 Ron Randall 07-Mar-1983  
62 0061 1 Global edit of all modules. Updated module names, idents,  
63 0062 1 copyright dates. Changed require files to BLISS library.  
64 0063 1 --  
65 0064 1

## Module Level Declarations

I 4  
16-Sep-1984 00:27:05  
14-Sep-1984 13:06:11  
VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]FCIMRA.BLI;1Page 3  
(3)FIG  
V04

```
: 67      0065 1 %SBTTL 'Module Level Declarations'  
68      0066 1 !  
69      0067 1 ! INCLUDE FILES:  
70      0068 1 !  
71      0069 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library  
72      0070 1 REQUIRE 'REQ:RNODEF';             ! RUNOFF variant definitions  
73      0201 1 !  
74      U 0202 1 %IF DSRPLUS %THEN  
75      U 0203 1 LIBRARY 'REQ:DPLLIB';          ! DSRPLUS BLISS Library  
76      U 0204 1 %ELSE  
77      U 0205 1 LIBRARY 'REQ:DSRLIB';           ! DSR BLISS Library  
78      U 0206 1 %FI  
79      U 0207 1 !  
80      U 0208 1 !  
81      U 0209 1 ! EXTERNAL REFERENCES:  
82      U 0210 1 !  
83      U 0211 1 EXTERNAL  
84      U 0212 1   fnct : fnct_definition,  
85      U 0213 1   mra  : REF FIXED_STRING,  
86      U 0214 1   pdt  : REF pdt_definition,  
87      U 0215 1   sca  : sca_definition,  
88      U 0216 1   tsf  : tsf_definition;  
89      U 0217 1 !  
90      U 0218 1 %IF DSRPLUS %THEN  
91      U 0219 1 EXTERNAL  
92      U 0220 1   entity_in_footnote,  
93      U 0221 1   l_character,  
94      U 0222 1   r_character;  
95      U 0223 1 !  
96      U 0224 1 EXTERNAL ROUTINE  
97      U 0225 1   foonum;  
98      U 0226 1 %FI  
99      U 0227 1 !  
100     U 0228 1 EXTERNAL ROUTINE  
101     U 0229 1   gcpus,  
102     U 0230 1   gcskip,  
103     U 0231 1   gtpc,  
104     U 0232 1   negind,  
105     U 0233 1   xmarg;  
106     U 0234 1 !
```

```
: 108 0235 1 GLOBAL ROUTINE fcimra : NOVALUE =
109 0236 1
110 0237 1 |+++
111 0238 1 | FUNCTIONAL DESCRIPTION:
112 0239 1
113 0240 1 | This routine is called prior to putting the first text character
114 0241 1 | into the MRA. It generates sufficient spaces to take care of
115 0242 1 | the left margin, takes care of paragraph indentation, and also
116 0243 1 | line spacing.
117 0244 1 | Prior to the call to FCIMRA, commands such as .SKIP, .TEST PAGE,
118 0245 1 | .LEFT MARGIN, .INDENT, and others have saved certain carriage
119 0246 1 | control information. FCIMRA finishes processing this information
120 0247 1 | before the first actual text character gets put into the MRA.
121 0248 1
122 0249 1 | FORMAL PARAMETERS: None
123 0250 1
124 0251 1 | IMPLICIT INPUTS: None
125 0252 1
126 0253 1 | IMPLICIT OUTPUTS: None
127 0254 1
128 0255 1 | ROUTINE VALUE:
129 0256 1 | COMPLETION CODES: None
130 0257 1
131 0258 1 | SIDE EFFECTS: None
132 0259 1 |---
133 0260 1
134 0261 2 | BEGIN
135 0262 2 | LOCAL
136 0263 2 | left_margin; ! Temporary computation of left margin.
137 0264 2
138 U 0265 2 |%IF DSRPLUS %THEN
139 U 0266 2
140 U 0267 2 | Exit if we are in FCIMRA as a result of resolving an entity.
141 U 0268 2
142 U 0269 2 | IF .entity_in_footnote
143 U 0270 2 | THEN
144 U 0271 2 | RETURN;
145 U 0272 2 |%FI
146 U 0273 2
147 U 0274 2 | left_margin = 0;
148 U 0275 2
149 U 0276 2 | IF .sca_para_pnd
150 U 0277 2 | THEN
151 U 0278 2 | Start a paragraph.
152 U 0279 2
153 U 0280 2
154 U 0281 2 | BEGIN
155 U 0282 2 | LOCAL
156 U 0283 2 | skip,
157 U 0284 2 | test_page;
158 U 0285 2
159 U 0286 2 | IF .pdt_skip LSS 0
160 U 0287 2 | THEN
161 U 0288 2 | skip = (.sca_spacing + 1) / 2
162 U 0289 2 | ELSE
163 U 0290 2 | skip = .pdt_skip * .sca_spacing;
164 U 0291 2
```

```
165      0292 3      test_page = (.pdt_tp + 1) * .sca_spacing + .skip;
166      0293
167      0294      IF .test_page NEQ 0
168      0295      THEN
169      0296          | Code for a test page command
170      0297          gtpc (.test_page);
171      0298
172      0299
173      0300
174      0301      IF .pdt_skip NEQ 0
175      0302      THEN
176      0303          | Code for skipping lines.
177      0304
178      0305
179      0306 4      BEGIN
180      0307
181      0308 4      IF .pdt_skip LSS 0
182      0309 4      THEN
183      0310 4          gcpo (.pdt_skip * .sca_spacing)
184      0311 4      ELSE
185      0312 4          gcskip (.pdt_skip * .sca_spacing);
186      0313 4
187      0314 3      END;
188      0315 3
189      0316 3      END
190      0317 2
191      0318 2      ELSE
192      0319 2          | Not a paragraph: just a new line.
193      0320 2
194      0321 2      IF .sca_spacing GTR 1
195      0322 2      THEN
196      0323 2          | Code for skipping between lines.
197      0324 2          gcskip (.sca_spacing - 1);
198      0325 2
199      0326 2      left_margin = .sca_lm + .sca_indent;
200      0327 2
201      0328 2
202      0329 2
203      0330 2      IF .left_margin LSS 0
204      0331 2      THEN
205      0332 3          BEGIN
206      0333 3              negind ();
207      0334 3              left_margin = 0;
208      0335 2          END;
209      0336 2
210      0337 2      IF .left_margin GEQ .sca_rm
211      0338 2      THEN
212      0339 3          BEGIN
213      0340 3              xmarg ();
214      0341 3              left_margin = 0;
215      0342 2          END;
216      0343 2
217      U 0344 2      %IF DSRPLUS %THEN
218      U 0345 2
219      U 0346 2      | Write a blank and the footnote number on the first line of a footnote.
220      U 0347 2
221      U 0348 2      IF .fnct_numbering AND .fnct_collecting AND (.fnct_first_line EQL 1)
```

```
222 U 0349 2 THEN
223 U 0350 2 BEGIN
224 U 0351 2
225 U 0352 2 | Reset flag that caused this action.
226 U 0353 2
227 U 0354 2 | fnct_first_line = 0;
228 U 0355 2
229 U 0356 2 | Set up the footnote number digits.
230 U 0357 2
231 U 0358 2 | foonum ();
232 U 0359 2
233 U 0360 2
234 U 0361 2 | Write out usual spaces plus one.
235 U 0362 2
236 U 0363 2 | INCR I FROM 1 TO (.left_margin + 1) DO
237 U 0364 2 | fs_wchar (mra, %C' ');
238 U 0365 2
239 U 0366 2
240 U 0367 2 | Write out the footnote number in brackets.
241 U 0368 2
242 U 0369 2 | fs_wchar (mra, %C '[');
243 U 0370 2
244 U 0371 2
245 U 0372 2 | Write the left character only if it is not 0.
246 U 0373 2
247 U 0374 2 | IF .fnct_number_l NEQ 0
248 U 0375 2 THEN
249 U 0376 2 | fs_wchar (mra, .l_character);
250 U 0377 2
251 U 0378 2 | fs_wchar (mra, .r_character);
252 U 0379 2 | fs_wchar (mra, %C ']');
253 U 0380 2 | fs_wchar (mra, %C ' ');
254 U 0381 2
255 U 0382 2
256 U 0383 2 | Write an extra space if no left character was written.
257 U 0384 2
258 U 0385 2 | IF .fnct_number_l EQL 0
259 U 0386 2 THEN
260 U 0387 2 | fs_wchar (mra, %C ' ');
261 U 0388 2
262 U 0389 2
263 U 0390 2 | Adjust character counters appropriately.
264 U 0391 2
265 U 0392 2 | tsf_int_hl = .left_margin + 6;
266 U 0393 2 | tsf_ext_hl = .left_margin + 6;
267 U 0394 2
268 U 0395 2 | Set in the left margin by 6 for subsequent footnote lines.
269 U 0396 2
270 U 0397 2 | sca_lm = .sca_lm + 6;
271 U 0398 2 END
272 U 0399 2 ELSE
273 0400 2
274 0401 2
275 0402 2 BEGIN
276 0403 2
277 0404 2 | INCR I FROM 1 TO .left_margin DO
278 0405 3 | fs_wchar (mra, %C '-' );
```

```

279      0406 3
280      0407 3
281      0408 3      tsf_int_nl = .left_margin;
282      U 0409 3      tsf_ext_nl = .left_margin;
283      U 0410 3      %IF DSRPLUS-%THEN
284      U 0411 3      sca_margin_pad = .left_margin;           ! Amount in the mra that was
285      U 0412 3      %FI                                         used for padding.
286      U 0413 2      END;
287      U 0414 2
288      U 0415 2      sca_indent = 0;
289      U 0416 2
290      U 0417 2      No paragraph pending.
291      U 0418 2
292      U 0419 2      sca_para_pnd = false;
293      U 0420 2
294      U 0421 2      Update word pointer.
295      U 0422 2
296      U 0423 2      sca_wrd_ptr = .fs_next (mra);
297      U 0424 2
298      U 0425 1      END;                                     ! End of FCIMRA

```

```

.TITLE FCIMRA
.IDENT \V04-000\

.EXTRN FNCT, MRA, PDT, SCA
.EXTRN TSF, GCPOS, GCSKIP
.EXTRN GTPC, NEGIND, XMARG

.PSECT SCODE$, NOWRT, 2

      007C 00000
      EF 9E 00002
      EF 9E 00009
      EF 9E 00010
      53 D4 00017
      64 D0 00019
      A4 E9 0001C
      65 D0 00020
      A0 D5 00023
      0C 18 00026
      62 D0 00028
      A1 9E 0002B
      02 C6 0002F
      08 11 00032
      62 D0 00034 1$:
      51 C5 00037
      01 C1 0003C 2$:
      51 C4 00041
      52 C0 00044
      09 13 00047
      50 DD 00049
      01 FB 0004B
      65 D0 00052 3$:
      50 D0 00055
      A0 D0 00059
      27 13 00059
      0E 18 0005B

.ENTRY FCIMRA, Save R2,R3,R4,R5,R6
MOVAB MRA, R6
MOVAB PDT, R5
MOVAB SCA+124, R4
CLRL LEFT_MARGIN
MOVL SCA+T24, R2
BLBC SCA+224, SS
MOVL PDT, R0
TSTL 4(R0)
BGEQ 1$
MOVL (R2), R1
MOVAB 1(R1), R2
DIVL2 #2, SKIP
BRB 2S
MOVL (R2), R1
MULL3 R1, 4(R0), SKIP
ADDL3 #1, 8(R0), R0
MULL2 R1, R0
ADDL2 SKIP, TEST_PAGE
BEQL 3S
PUSHL TEST PAGE
CALLS #1, GTPC
MOVL PDT, R0
MOVL 4(R0), R0
BEQL 7S
BGEQ 4S

```

0235  
0274  
0288  
0276  
0286  
0288  
0290  
0292  
0294  
0299  
0301  
0308

FIG  
V04-  
: R  
: S  
: L

## Module Level Declarations

N 4  
16-Sep-1984 00:27:05  
14-Sep-1984 13:06:11 VAX-11 Bliss-32 v4.0-742  
[RUNOFF.SRC]FCIMRA.BLI;1Page 8  
(4)

7E	00000000G	50	00	84	C5 0005D	MULL3	@SCA+124, R0, -(SP)	0310
		EF		01	FB 00062	CALLS	#1, GCPOS	
7E		50	00	17	11 00069	BRB	7\$	0312
				09	11 00070	MULL3	@SCA+124, R0, -(SP)	
		01		62	D1 00072	BRB	6\$	0321
7E	00000000G	62		0B	15 00075	CMPL	(R2), #1	0326
		EF		01	C3 00077	SUBL3	#1, (R2), -(SP)	
53	F8	B4	60	01	FB 0007B	CALLS	#1, GCSKIP	0328
				09	C1 00082	ADDL3	SCA+220, @SCA+116, LEFT_MARGIN	
	00000000G	EF		09	18 00088	BGEQ	8\$	0330
				00	FB 0008A	CALLS	#0, NEGIND	0333
		FC	B4	53	D4 00091	CLRL	LEFT_MARGIN	0334
				53	D1 00093	CMPL	LEFT_MARGIN, @SCA+120	0337
	00000000G	EF		09	19 00097	BLSS	9\$	0340
				00	FB 00099	CALLS	#0, XMARG	0341
				53	D4 000A0	CLRL	LEFT_MARGIN	
		50		66	D0 000A2	MOVL	MRA, R0	0405
				51	D0 000A5	MOVL	MRA, R1	
				52	D4 000A8	CLRL	I	
			04	0A	11 000AA	BRB	11\$	
			B0	20	90 000AC	MOVBL	#32, @4(R0)	
				04	A0 D6 000B0	INCL	4(R0)	
				0C	A1 D6 000B3	INCL	12(R1)	
F2		52	00000000G	53	F3 000B6	AOBLEQ	LEFT_MARGIN, I, 10\$	0404
				51	EF D0 000BA	MOVL	TSF, R1	0405
		61		53	D0 000C1	MOVL	LEFT_MARGIN, (R1)	0407
		04	A1	53	D0 000C4	MOVL	LEFT_MARGIN, 4(R1)	0408
				7C	A4 7C 000C8	CLRQ	SCA+220	0415
				A4	DO 000CB	MOVL	4(R0), SCA+248	0423
				04	000D0	RET		
				04	000D0			0425

; Routine Size: 209 bytes, Routine Base: \$CODE\$ + 0000

; 299 0426 1  
; 300 0427 1 END  
; 301 0428 0 ELUDOM

! End of module

## PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	209	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

## Library Statistics

File	----- Symbols -----	Pages Mapped	Processing Time
	Total    Loaded    Percent		

FCIMRA  
V04-000

Module Level Declarations

B 5  
16-Sep-1984 00:27:05  
14-Sep-1984 13:06:11  
VAX-11 Bliss-32 V4.0-742  
[RUNOFF.SRC]FCIMRA.BLI;1

Page 9  
(4)

\*\*F

: \$255\$DUA28:[SYSLIB]XPORT.L32;1 590 0 252  
: -\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1 1248 0 2 86 00:00.1  
: 00:00.2

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:FCIMRA/OBJ=OBJ\$:FCIMRA MSRC\$:FCIMRA/UPDATE=(ENH\$:FCIMRA)

: Size: 209 code + 0 data bytes  
: Run Time: 00:06.1  
: Elapsed Time: 00:18.1  
: Lines/CPU Min: 4230  
: Lexemes/CPU-Min: 15924  
: Memory Used: 75 pages  
: Compilation Complete

0341 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

