



```

FFFFFFFFF  CCCCCCCC  IIIIII  MM      MM  RRRRRRRR  AAAAAA
FFFFFFFFF  CCCCCCCC  IIIIII  MM      MM  RRRRRRRR  AAAAAA
FF        CC        II      MMMM  MMMM  RR      RR  AA      AA
FF        CC        II      MMMM  MMMM  RR      RR  AA      AA
FF        CC        II      MM   MM   RR      RR  AA      AA
FF        CC        II      MM   MM   RR      RR  AA      AA
FFFFFFFFF  CC        II      MM   MM   RRRRRRRR  AA      AA
FFFFFFFFF  CC        II      MM   MM   RRRRRRRR  AA      AA
FF        CC        II      MM   MM   RR  RR  AAAAAAAAAA
FF        CC        II      MM   MM   RR  RR  AAAAAAAAAA
FF        CC        II      MM   MM   RR      RR  AA      AA
FF        CC        II      MM   MM   RR      RR  AA      AA
FF        CCCCCCCC  IIIIII  MM   MM   RR      RR  AA      AA
FF        CCCCCCCC  IIIIII  MM   MM   RR      RR  AA      AA

```

```

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 MODULE fcimra ( IDENT = 'V04-000'
2 P 0002 0   %BLISS32[, ADDRESSING_MODE (EXTERNAL   = LONG_RELATIVE,
3   0003 0   NONEXTERNAL = LONG_RELATIVE)]
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:      DSR (Digital Standard RUNOFF) / DSRPLUS
33  0033 1
34  0034 1 ABSTRACT:
35  0035 1
36  0036 1     Special handling for first character written into the MRA.
37  0037 1
38  0038 1 ENVIRONMENT:  Transportable
39  0039 1
40  0040 1 AUTHOR:      R.W.Friday      CREATION DATE:  June, 1978
41  0041 1

```

Revision History

```
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

```

: 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      0204 1 %ELSE
: 77      0205 1 LIBRARY 'REQ:DSRLIB';           ! DSR BLISS Library
: 78      0206 1 %FI
: 79      0207 1
: 80      0208 1
: 81      0209 1 ! EXTERNAL REFERENCES:
: 82      0210 1
: 83      0211 1 EXTERNAL
: 84      0212 1     fnct : fnct_definition,
: 85      0213 1     mra  : REF FIXED_STRING,
: 86      0214 1     pdt  : REF pdt_definition,
: 87      0215 1     sca  : sca_definition,
: 88      0216 1     tsf  : tsf_definition;
: 89      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      0226 1 %FI
: 99      0227 1
: 100     0228 1 EXTERNAL ROUTINE
: 101     0229 1     gcpos,
: 102     0230 1     gcskip,
: 103     0231 1     gtpc,
: 104     0232 1     negind,
: 105     0233 1     xmarg;
: 106     0234 1

```

```

108 0235 1 GLOBAL ROUTINE fcimra : NOVALUE =
109 0236 1
110 0237 1
111 0238 1 ++
112 C239 1 FUNCTIONAL DESCRIPTION:
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 0273 2
147 0274 2 left_margin = 0;
148 0275 2
149 0276 2 IF .sca_para_pnd
150 0277 2 THEN
151 0278 2 |
152 0279 2 | Start a paragraph.
153 0280 2 |
154 0281 2 BEGIN
155 0282 2 LOCAL
156 0283 2 skip,
157 0284 2 test_page;
158 0285 2
159 0286 2 IF .pdt_skip LSS 0
160 0287 2 THEN
161 0288 2 skip = (.sca_spacing + 1) / 2
162 0289 2 ELSE
163 0290 2 skip = .pdt_skip * .sca_spacing;
164 0291 2

```

Module Level Declarations

```

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

```

```

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

```



Module Level Declarations

```
: 279 0406 3
: 280 0407      tsf_int_nl = .left_margin;
: 281 0408      tsf_ext_hl = .left_margin;
: 282 U 0409 %IF DSRPLUS %THEN
: 283 U 0410 sca_margin_pad = .left_margin;
: 284 U 0411
: 285 U 0412 %FI
: 286 U 0413      END;
: 287 U 0414
: 288 U 0415      sca_indent = 0;
: 289 U 0416      !
: 290 U 0417      ! No paragraph pending.
: 291 U 0418      !
: 292 U 0419      sca_para_pnd = false;
: 293 U 0420      !
: 294 U 0421      ! Update word pointer.
: 295 U 0422      !
: 296 U 0423      sca_wrd_pntr = .fs_next (mra);
: 297 U 0424      !
: 298 U 0425      END;
```

! Amount in the mra that was  
! used for padding.

! End of FCIMRA

```
.TITLE FCIMRA
.IDENT \V04-000\

.EXTRN FNCT, MRA, PDT, SCA
.EXTRN TSF, GCPOS, GC SKIP
.EXTRN GTPC, NEGIND, XMARG
```

```
.PSECT $CODE$,NOWRT,2
```

```
.ENTRY FCIMRA, Save R2,R3,R4,R5,R6
```

```
56 00000000G EF 9E 00002 007C 00000
55 00000000G EF 9E 00009
54 00000000G EF 9E 00010
53 04 00017
52 64 00019
52 64 A4 E9 0001C
50 04 00020
0C 04 A0 D5 00023
0C 18 00026
51 01 62 D0 00028
52 01 A1 9E 0002B
52 02 C6 0002F
08 11 00032
51 51 62 D0 00034 1$:
50 04 A0 51 C5 00037 2$:
50 08 A0 01 C1 0003C 2$:
50 51 C4 00041
50 52 C0 00044
09 13 00047
50 DD 00049
00000000G EF 01 FB 0004B 3$:
50 65 D0 00052 3$:
50 04 A0 D0 00055 3$:
27 13 00059
0E 18 0005B
```

```
MOVAB MRA, R6
MOVAB PDT, R5
MOVAB SCA+124, R4
CLRL LEFT MARGIN
MOVL SCA+T24, R2
BLBC SCA+224, 5$
MOVL PDT, R0
TSTL 4(R0)
BGEQ 1$
MOVL (R2), R1
MOVAB 1(R1), R2
DIVL2 #2, SKIP
BRB 2$
MOVL (R2), R1
MULL3 R1, 4(R0), SKIP
ADDL3 #1, 8(R0), R0
MULL2 R1, R0
ADDL2 SKIP, TEST_PAGE
BEQL 3$
PUSHL TEST_PAGE
CALLS #1, GTPC
MOVL PDT, R0
MOVL 4(R0), R0
BEQL 7$
BGEQ 4$
```

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

```
: R
:
:
:
:
:
:
:
:
:
:
:
:
: S
: R
: E
: L
```

Module Level Declarations

7E	00000000G	50	00	B4	C5	0005D	MULL3	@SCA+124, R0, -(SP)	0310	
		EF		01	FB	00062	CALLS	#1, GCPOS		
				17	11	00069	BRB	7\$		
7E		50	00	B4	C5	0006B	MULL3	@SCA+124, R0, -(SP)	0312	
				09	11	00070	BRB	6\$		
				01	62	D1	00072	5\$: CMPL	(R2), #1	0321
				08	15	00075	BLEQ	7\$		
7E	00000000G	62		01	C3	00077	SUBL3	#1, (R2), -(SP)	0326	
		EF		01	FB	0007B	CALLS	#1, GCSKIP		
53	F8	B4	60	A4	C1	00082	7\$: ADDL3	SCA+220, @SCA+116, LEFT_MARGIN	0328	
				09	18	00088	BGEQ	8\$	0330	
	00000000G	EF		00	FB	0008A	CALLS	#0, NEGIND	0333	
				53	D4	00091	CLRL	LEFT_MARGIN	0334	
	FC	B4		53	D1	00093	8\$: CMPL	LEFT_MARGIN, @SCA+120	0337	
				09	19	00097	BLSS	9\$		
	00000000G	EF		00	FB	00099	CALLS	#0, XMARG	0340	
				53	D4	000A0	CLRL	LEFT_MARGIN	0341	
				50	66	D0	000A2	9\$: MOVL	MRA, R0	0405
				51	66	DU	000A5	MOVL	MRA, R1	
				52	D4	000A8	CLRL	I		
				0A	11	000AA	BRB	11\$		
	04	B0		20	90	000AC	10\$: MOVB	#32, @4(R0)		
			04	A0	D6	000B0	INCL	4(R0)		
			0C	A1	D6	000B3	INCL	12(R1)		
F2		52		53	F3	000B6	11\$: AOBLEQ	LEFT_MARGIN, I, 10\$	0404	
		51	00000000G	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	
				60	A4	7C	000C8	CLRQ	SCA+220	0415
	7C	A4		04	A0	D0	000CB	MOVL	4(R0), SCA+248	0423
				04	000DU		RET		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		

```

:
:  $255$DUA28:[SYSLIB]XPORT.L32;1          590      0      0      252      00:00.1
:  $255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1    1248     25     2      86      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


FIND LIS

ENDWRD LIS

ERROR LIS

FIGURE LIS

FLGSEM LIS

FOOFIL LIS

GCODE LIS

FOCMRA LIS

FNONLY LIS

FUNFNJ LIS

FOOBOT LIS

GBLDCL LIS

FNDPLG LIS

FOOOUT LIS

FORMAT LIS