

```
RRRRRRRRRRRR  UUU      UUU  NNN      NNN      00000000  FFFFFFFFFFFFFFFF  FFFFFFFFFFFFFFFF
RRRRRRRRRRRR  UUU      UUU  NNN      NNN      00000000  FFFFFFFFFFFFFFFF  FFFFFFFFFFFFFFFF
RRRRRRRRRRRR  UUU      UUU  NNN      NNN      00000000  FFFFFFFFFFFFFFFF  FFFFFFFFFFFFFFFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRRRRRRRRRRR  UUU      UUU  NNN      NNN      000          000  FFFFFFFFFFFFFFFF  FFFFFFFFFFFFFFFF
RRRRRRRRRRRR  UUU      UUU  NNN      NNN      000          000  FFFFFFFFFFFFFFFF  FFFFFFFFFFFFFFFF
RRRRRRRRRRRR  UUU      UUU  NNN      NNN      000          000  FFFFFFFFFFFFFFFF  FFFFFFFFFFFFFFFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUU      UUU  NNN      NNN      000          000  FFF          FFF
RRR          RRR  UUUUUUUUUUUUUUU  NNN      NNN      00000000  FFF          FFF
RRR          RRR  UUUUUUUUUUUUUUU  NNN      NNN      00000000  FFF          FFF
RRR          RRR  UUUUUUUUUUUUUUU  NNN      NNN      00000000  FFF          FFF
```

Sym

NDX

NDX

NUM

NUM

OUT

PAC

PAC

PAC

PAC

PAC

PAC

PAC

PAC

PAD

PAG

PAG

PAG

PAG

PAG

PAG

PER

PUT

RCO

RIN

RLI

RNO

RNO

RTY

SAV

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

STR

```

FFFFFFFFF 000000 000000 000000 UU UU TTTTTTTTTT
FFFFFFFFF 000000 000000 000000 UU UU TTTTTTTTTT
FF 00 00 00 00 00 00 UU UU TT
FF 00 00 00 00 00 00 UU UU TT
FF 00 00 00 00 00 00 UU UU TT
FF 00 00 00 00 00 00 UU UU TT
FFFFFFF 00 00 00 00 00 00 UU UU TT
FFFFFFF 00 00 00 00 00 00 UU UU TT
FF 00 00 00 00 00 00 UU UU TT
FF 00 00 00 00 00 00 UU UU TT
FF 00 00 00 00 00 00 UU UU TT
FF 00 00 00 00 00 00 UU UU TT
FF 000000 000000 000000 UUUUUUUUUU TT
FF 000000 000000 000000 UUUUUUUUUU TT

```

```

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

```

```

LL 111111 SSSSSSSS
LL 111111 SSSSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SSSSSS
LL 11 SSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS

```

```

1 0001 0 %TITLE 'performs outputting of footnotes'
2 0002 0 MODULE fooout ( IDENT = 'v04-000'
3 P 0003 0 %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
4 0004 0 NONEXTERNAL = LONG_RELATIVE)]
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
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: Puts out footnotes.
35 0035 1
36 0036 1 ENVIRONMENT: Transportable
37 0037 1
38 0038 1 AUTHOR: R.W.Friday CREATION DATE: September, 1978
39 0039 1

```

```

: 41      0040 1 %SBTTL 'Revision History'
: 42      0041 1  MODIFIED BY:
: 43      0042 1
: 44      0043 1
: 45      0044 1      012      RER00012      Ron Randall      13-Apr-1983
: 46      0045 1      Fixed obscure bug with numbered footnotes.
: 47      0046 1
: 48      0047 1      011      RER00011      Ron Randall      06-Apr-1983
: 49      0048 1      For DSRPLUS:
: 50      0049 1      For numbered footnotes, a line of dashes is output first.
: 51      0050 1      010      RER00010      Ron Randall      07-Mar-1983
: 52      0051 1      Global edit of all modules. Updated module names, idents,
: 53      0052 1      copyright dates. Changed require files to BLISS library.
: 54      0053 1      --
: 55      0054 1
```

```
57 0055 1 %SBTTL 'Module Level Declarations'
58 0056 1
59 0057 1 : INCLUDE FILES:
60 0058 1
61 0059 1 LIBRARY 'NXPORT:XPORT'; : XPORT Library
62 0060 1 REQUIRE 'REQ:RNODEF'; : RUNOFF variant definitions
63 0191 1
64 U 0192 1 %IF DSRPLUS %THEN
65 U 0193 1 LIBRARY 'REQ:DPLLIB'; : DSRPLUS BLISS Library
66 0194 1 %ELSE
67 0195 1 LIBRARY 'REQ:DSRLIB'; : DSR BLISS Library
68 0196 1 %FI
69 0197 1
70 0198 1
71 0199 1 : EXTERNAL REFERENCES:
72 0200 1
73 0201 1 EXTERNAL LITERAL
74 0202 1 RINTES : UNSIGNED (8);
75 0203 1
76 0204 1 EXTERNAL LITERAL
77 0205 1 RNFILE; :Error messages
78 0206 1
79 0207 1 EXTERNAL
80 0208 1 FNCT : FNCT_DEFINITION,
81 0209 1 FNISIZ : FN_INT_SIZE_DEFINITION,
82 0210 1 FOOREC : FOOREC_DEFINITION,
83 0211 1 GCA : GCA_DEFINITION,
84 0212 1 MRA : REF_FIXED_STRING,
85 0213 1 PHAN : PHAN_DEFINITION,
86 0214 1 TSF : TSF_DEFINITION;
87 0215 1
88 0216 1 EXTERNAL
89 0217 1 FOOMRA : FIXED_STRING, :Substitute MRA.
90 0218 1 FOOTSF : VECTOR [TSF_SIZE]; :Substitute TSF area
91 0219 1
92 0220 1 EXTERNAL
93 0221 1 FOHMRA,
94 0222 1 FOHTSF;
95 0223 1
96 U 0224 1 %IF DSRPLUS %THEN
97 U 0225 1 EXTERNAL
98 U 0226 1 FRA : FIXED_STRING;
99 U 0227 1
100 U 0228 1 EXTERNAL ROUTINE
101 U 0229 1 LOUT;
102 0230 1 %FI
103 0231 1
104 0232 1 EXTERNAL ROUTINE
105 0233 1 ERMS,
106 0234 1 FOOFIL,
107 0235 1 OUTLIN,
108 0236 1 OUTPAS;
109 0237 1
```

```

111 0238 1 %SBTTL 'body of routine'
112 0239 1 GLOBAL ROUTINE fooout (OUTPUT_COUNT) : NOVALUE =
113 0240 1
114 0241 1 **
115 0242 1 FUNCTIONAL DESCRIPTION:
116 0243 1
117 0244 1     FOOOUT outputs as many footnotes as it is requested to do. For each
118 0245 1     footnote, all it does is read the appropriate number of TSF/MRA pairs
119 0246 1     from the footnote file, and calls OUTLIN to process the information.
120 0247 1     The number of TSF/MRA sets for any particular footnote is stored at
121 0248 1     FNISIZ [n].
122 0249 1
123 0250 1     FOOOUT also takes care of index entries that it finds. The transaction
124 0251 1     numbers are remapped into a different set, so that they get merged with
125 0252 1     entries from the main body of the document.
126 0253 1
127 0254 1     After FOOOUT has output the footnotes it's supposed to output, it
128 0255 1     leaves the file open. It does not close the file, or attempt to clean
129 0256 1     it up in any way.
130 0257 1
131 0258 1 FORMAL PARAMETERS:
132 0259 1
133 0260 1     OUTPUT_COUNT - Number of footnotes to be output.
134 0261 1
135 0262 1 IMPLICIT INPUTS:      None
136 0263 1
137 0264 1 IMPLICIT OUTPUTS:    None
138 0265 1
139 0266 1 ROUTINE VALUE:
140 0267 1 COMPLETION CODES:    None
141 0268 1
142 0269 1 SIDE EFFECTS:        None
143 0270 1 --
144 0271 1
145 0272 2     BEGIN
146 0273 2     LOCAL
147 0274 2     COUNT,
148 0275 2     FOO_FIRST_XTN,
149 0276 2     HOLD_PAGING,
150 0277 2     MAP_FIRST_XTN,
151 0278 2     STATUS;
152 0279 2
153 0280 2     IF .OUTPUT_COUNT LEQ 0      ! If this is a meaningless call,
154 0281 2     THEN
155 0282 2     RETURN                      ! ignore it.
156 0283 2     ELSE                          ! Otherwise, check the validity of the request
157 0284 3     BEGIN
158 0285 3
159 0286 3     IF .OUTPUT_COUNT GTR .FNCT_READY ! If the user asked for more
160 0287 3     THEN                          ! footnotes to be output than are
161 0288 4     BEGIN                          ! ready to be output,
162 0289 4     ERMS (RNFILE, CH$PTR (UPLIT ('FOOOUT'), 6); ! report the error
163 0290 4     COUNT = .FNCT_READY;                ! and use the real count instead.
164 0291 4     END
165 0292 3     ELSE
166 0293 3     COUNT = .OUTPUT_COUNT;            ! Otherwise, setup to output the #
167 0294 3     the user requested.

```

```

168      0295      2      END;
169      0296      2
170      0297      2      ! NOTE: The change to not using pool for footnotes necessitated changing a
171      0298      2      ! test for .FNPOOL. This change MAY have reintroduced an old bug from
172      0299      2      ! either # 020 or # 025. Check for it and beware.
173      0300      2      ! RUNOFF V2(1), FN IDENT 008; 9-2-1980. kad.
174      0301      2
175      0302      2      IF FOOFIL (FOO_OPIN) NEQ FOO_NORMAL
176      0303      2      THEN
177      0304      2          BEGIN ! Can't read back footnotes: give up and forget them.
178      0305      2          FOOFIL (FOO_CLOS);
179      0306      2          FOOFIL (FOO_DELE);
180      0307      2          FNCT_COUNT = 0;
181      0308      2          FNCT_READY = 0;
182      0309      2          FNCT_WAITING = 0;
183      0310      2          FNCT_OLD = 0;
184      0311      2          RETURN
185      0312      2          END;
186      0313      2
187      0314      2      !The footnote file was opened successfully.
188      0315      2      FNCT_EXPANDING = TRUE; !Signal the fact that footnotes are being expanded.
189      0316      2      HOLD_PAGING = .PHAN_PAGING; !Go into non-paging mode to
190      0317      2      PHAN_PAGING = FALSE; !avoid spurious page breaks.
191      0318      2      !Use the footnote working areas to save results in.
192      0319      2      FOHTSF = .TSF; !Save current status.
193      0320      2      FOHMRA = .MRA;
194      0321      2      TSF = FOOTSF; !Switch to other working area.
195      0322      2      MRA = FOO_MRA;
196      0323      2      FOO_FIRST_XTN = 0; !Initialize transaction number
197      0324      2      MAP_FIRST_XTN = 0;
198      0325      2
199      U 0326      2      %IF DSRPLUS %THEN
200      UU 0327      2      !
201      UU 0328      2      ! If outputting numbered footnotes, first output a line of dashes.
202      UU 0329      2      !
203      UU 0330      2      IF .FNCT_NUMBERING
204      UU 0331      2      THEN
205      UU 0332      2          BEGIN
206      UU 0333      2          !
207      UU 0334      2          ! Initialize fixed string.
208      UU 0335      2          !
209      UU 0336      2          FS_INIT (MRA);
210      UU 0337      2          !
211      UU 0338      2          ! Write dashes into the fixed string.
212      UU 0339      2          !
213      UU 0340      2          INCR I FROM 1 TO 15 DO
214      UU 0341      2              FS_WCHAR (MRA, %C'-');
215      UU 0342      2          !
216      UU 0343      2          !
217      UU 0344      2          ! Update the appropriate counts.
218      UU 0345      2          !
219      UU 0346      2          TSF_INT_HL = 15;
220      UU 0347      2          TSF_EXT_HL = 15;
221      UU 0348      2          TSF_INT_VL = 0;
222      UU 0349      2          !
223      UU 0350      2          ! Put out the line as OUTLIN does.
224      UU 0351      2          !

```

```
225 U 0352 2 FS_INIT (FRA);
226 U 0353 2 GCA_LINE_PEND = 0;
227 U 0354 2 LOUT ();
228 U 0355 2 END;
229 U 0356 2 %FI
230 U 0357 2
231 U 0358 2 !Now actually read back the records.
232 U 0359 2 INCR I FROM 0 TO (.COUNT - 1) DO
233 U 0360 2 INCR J FROM 1 TO .FNISIZ [.FNCT_OLD + .I] DO ! (J is a dummy counter, not really used anyw
234 U 0361 2 BEGIN
235 U 0362 2 !Determine which kind of record is coming: a TSF/MRA pair, or a 'pass through'
236 U 0363 2 !record.
237 U 0364 2 STATUS = FOOFIL (FOO_RHDR);
238 U 0365 2
239 U 0366 2 IF .STATUS NEQ FOO_NORMAL
240 U 0367 2 THEN
241 U 0368 2 !Exit loop if record can't be read.
242 U 0369 2 EXITLOOP;
243 U 0370 2
244 U 0371 2 !Is what's coming a TSF/MRA pair, or is it 'pass through' information?
245 U 0372 2 IF .FOOREC_MAJOR_TYPE EQL FOOREC_MAJ_PASS
246 U 0373 2 THEN
247 U 0374 2 !Direct 'pass through' information to the appropriate destination
248 U 0375 2 BEGIN
249 U 0376 2 !First read the 'pass through' information
250 U 0377 2 STATUS = FOOFIL (FOO_PREAD);
251 U 0378 2
252 U 0379 2 !Make sure the read worked.
253 U 0380 2 IF .STATUS NEQ FOO_NORMAL
254 U 0381 2 THEN
255 U 0382 2 !Information couldn't be read. !Quit processing.
256 U 0383 2 EXITLOOP;
257 U 0384 2
258 U 0385 2 !Map transaction number, if any.
259 U 0386 2 IF .FOOREC_XTN_PTR GTR 0
260 U 0387 2 THEN
261 U 0388 2 !This pass thru record contains a transaction number. Remap it into something else.
262 U 0389 2 BEGIN
263 U 0390 2 BIND
264 U 0391 2 PASS_THRU_RECORD = .FOOREC_ADDRESS : VECTOR;
265 U 0392 2 !Note: Never change transaction numbers that are zero.
266 U 0393 2
267 U 0394 2 IF .PASS_THRU_RECORD [.FOOREC_XTN_PTR - 1] NEQ 0
268 U 0395 2 THEN
269 U 0396 2 BEGIN
270 U 0397 2
271 U 0398 2 IF .FOO_FIRST_XTN EQL 0
272 U 0399 2 THEN
273 U 0400 2 !Establish start of mapping from temporarily assigned
274 U 0401 2 !transaction numbers to the normal set.
275 U 0402 2 BEGIN
276 U 0403 2 MAP_FIRST_XTN = .GCA_NORMAL_XTN; !Establish Y(1).
277 U 0404 2 FOO_FIRST_XTN = .PASS_THRU_RECORD [.FOOREC_XTN_PTR - 1]; !Establish X(1).
278 U 0405 2 END;
279 U 0406 2
280 U 0407 2 !Translate the temporary transaction number assigned
281 U 0408 2 !to this pass thru record.
```



```
282 0409 6 !This is done by simply substituting the current Y(1)
283 0410 6 ! (which is GCA_NORMAL_XTN) for the given X(1).
284 0411 6 ! Then, GCA_NORMAL_XTN is bumped, thereby giving a
285 0412 6 ! new Y(1) for the next such record, if any.
286 0413 6 PASS THRU RECORD [.FOOREC_XTN_PTR - 1] = .GCA_NORMAL_XTN;
287 0414 6 GCA_NORMAL_XTN = .GCA_NORMAL_XTN + 1;
288 0415 6 ENC
289 0416 6
290 0417 4 END;
291 0418 4
292 0419 4 OUTPAS (.FOOREC_RECORD_SIZE, .FOOREC_ADDRESS, .FOOREC_XTN_PTR,
293 0420 4 .FOOREC_MINOR_TYPE)
294 0421 4 END
295 0422 3 ELSE
296 0423 3 !It's TSF/MRA information to be processed, so get it and process it.
297 0424 4 BEGIN
298 0425 4 FOOFIL (FOO_READ);
299 0426 4
300 0427 4 !Make sure the read worked.
301 0428 4 IF .STATUS NEQ FOO_NORMAL
302 0429 4 THEN
303 0430 4 !Information couldn't be read. !Quit processing.
304 0431 4 EXITLOOP;
305 0432 4
306 0433 4 !See if this record has anything to do with indexing.
307 0434 4 !If so, the record either contains an entry for the index,
308 0435 4 !or else it contains text that refers to something in the
309 0436 4 !index (or the index refers to it, depending on your point of
310 0437 4 !view.
311 0438 4
312 0439 4 !Records containing index entries always occur before the
313 0440 4 !record to which they apply; each record describes one
314 0441 4 !entry. Further, each index entry has associated with it
315 0442 4 !a unique number, TSF_FIRST_XTN. These transaction numbers
316 0443 4 !are strictly increasing from record to record. Assume that
317 0444 4 !there are n such records, and let the transaction numbers
318 0445 4 !be designated by X(1), X(2), .... X(n).
319 0446 4
320 0447 4 !Records which refer to index entries come after the index
321 0448 4 !entry records; they are chained to the index entries by
322 0449 4 !having the transaction numbers recorded in the corresponding
323 0450 4 !TSF. TSF_FIRST_XTN will have the value X(1), and TSF_LAST_XTN
324 0451 4 !will have the value X(n).
325 0452 4
326 0453 4 !The problem is that X(1).....X(n) cannot be directly used
327 0454 4 !by the routine ASGXTN (which is in the module XTN). That
328 0455 4 !is because the association of transaction numbers to pages
329 0456 4 !requires that the transaction numbers passing through ASGXTN
330 0457 4 !be steadily increasing, with no gaps, FOR THE ENTIRE FILE.
331 0458 4 !Now, X(1).....X(n) are steadily increasing, but only
332 0459 4 !for the footnote. However, they can be mapped into the
333 0460 4 !"correct" set, which is accomplished here.
334 0461 4 !Note that the new set will be designated by Y(1).....Y(n).
335 0462 4 !Further, Y(1) is guaranteed to be GCA_NORMAL_XTN.
336 0463 4 IF .TSF_FIRST_XTN NEQ 0
337 0464 4 THEN
338 0465 4
```

```

339 0466 4 !This record has something to do with indexing.
340 0467 4 IF .TSF_INDEX
341 0468 4 THEN
342 0469 4 !Process index entry.
343 0470 5 BEGIN
344 0471 5
345 0472 5 IF .FOO_FIRST_XTN EQL 0
346 0473 5 THEN
347 0474 5 !Establish start of mapping from temporarily assigned
348 0475 5 !transaction numbers to the normal set.
349 0476 6 BEGIN
350 0477 6 MAP_FIRST_XTN = .GCA_NORMAL_XTN; !Establish Y(1).
351 0478 6 FOO_FIRST_XTN = .TSF_FIRST_XTN; !Establish X(1).
352 0479 5 END;
353 0480 5
354 0481 5 !Translate the temporary transaction number assigned
355 0482 5 !to this index entry into a normal one.
356 0483 5 !This is done by simply substituting the current Y(1)
357 0484 5 !(which is GCA_NORMAL_XTN) for the given X(1).
358 0485 5 !Then, GCA_NORMAL_XTN is bumped, thereby giving a
359 0486 5 !new Y(1) for the next index entry, if any.
360 0487 5 TSF_FIRST_XTN = .GCA_NORMAL_XTN;
361 0488 5 TSF_LAST_XTN = .GCA_NORMAL_XTN;
362 0489 5 GCA_NORMAL_XTN = .GCA_NORMAL_XTN + 1;
363 0490 5 END
364 0491 4 ELSE
365 0492 4 !Process reference to index entry.
366 0493 4 !Notice that earlier, the starting X(1) and Y(1) were
367 0494 4 !established. But that was for the start of the
368 0495 4 !footnote, not for the current record. In the meantime,
369 0496 4 !several sets of index entries and references may have
370 0497 4 !gone through here. And there is no way to separate them.
371 0498 4 !That means you can't simply use MAP_FIRST_XTN as the
372 0499 4 !new Y(1).
373 0500 5 BEGIN
374 0501 5 LOCAL
375 0502 5 XTN_OFFSET;
376 0503 5
377 0504 5 !Compute the gap between the X(1) on this record
378 0505 5 !and that at the start of the footnote processing.
379 0506 5 !XTN_OFFSET, the result, will be zero for the very first
380 0507 5 !record that comes through.
381 0508 5 XTN_OFFSET = .TSF_FIRST_XTN - .FOO_FIRST_XTN;
382 0509 5
383 0510 5 !Now, compute the desired Y(n).
384 0511 5 !The value MAP_FIRST_XTN+XTN_OFFSET gets you the Y(1)
385 0512 5 !you need. The value TSF_LAST_XTN - TSF_FIRST_XTN gets
386 0513 5 !you "n".
387 0514 5 TSF_LAST_XTN = .MAP_FIRST_XTN + .XTN_OFFSET + (.TSF_LAST_XTN - .TSF_FIRST_XTN);
388 0515 5
389 0516 5 !This is left as an exercise for the reader.
390 0517 5 TSF_FIRST_XTN = .MAP_FIRST_XTN + .XTN_OFFSET;
391 0518 5
392 0519 5 !As an additional exercise, prove that it's
393 0520 5 !not necessary to clear FOO_FIRST_XTN now, or later;
394 0521 5 !Further, show that it doesn't matter whether or
395 0522 5 !not you do clear it. Using the results of these

```

```

: 396      0523      5      !exercises simplify this block of code.
: 397      0524      4      END;
: 398      0525      4
: 399      0526      4      !Write out the record.
: 400      0527      4      OUTLIN (0);      !(Parameter is meaningless in this case).
: 401      0528      3      END;
: 402      0529      3
: 403      0530      2      END;
: 404      0531      2
: 405      0532      2      TSF = .FOHTSF;      !Restore interrupted status.
: 406      0533      2      MRA = .FOHMRA;
: 407      0534      2      PHAN_PAGING = .HOLD_PAGING;
: 408      0535      2      FNCT_READY = .FNCT_READY - .COUNT;      !Count off the footnotes that have been output.
: 409      0536      2      FNCT_COUNT = .FNCT_COUNT - .COUNT;      !Count off the footnotes that have been output.
: 410      0537      2      FNCT_OLD = .FNCT_OLD + .COUNT;      !Update count of obsolete footnotes.
: 411      0538      2      FNCT_EXPANDING = FALSE;      !No longer expanding footnotes
: 412      0539      2
: 413      0540      2      !Get rid of the work files if there are no more footnotes to be output.
: 414      0541      2      IF .FNCT_COUNT LEQ 0
: 415      0542      2      THEN
: 416      0543      2      !All footnotes have been output.
: 417      0544      3      BEGIN
: 418      0545      3      FOOFIL (FOO_CLOS);      !Close the file
: 419      0546      3      FOOFIL (FCO_DELE);      !and delete it.
: 420      0547      3      FNCT_OLD = 0;      !And clean up the tables a bit.
: 421      0548      3      FNCT_READY = 0;
: 422      0549      3      FNCT_WAITING = 0;
: 423      0550      2      END;
: 424      0551      2
: 425      0552      1      END;      !End of FOOOUT

```

```

          .TITLE FOOOUT performs outputting of footnotes
          .IDENT \V04-000\
          .PSECT $PLITS,NOWRT,NOEXE,2
00 00 54 55 4F 4F 4F 46 0000 P.AAA: .ASCII \FOOOUT\<0><0>
          .EXTRN RINTES, PNFIL, FNCT
          .EXTRN FNISIZ, FOOREC, GCA
          .EXTRN MRA, PHAN, TSF, FOMRA
          .EXTRN FOOTSF, FOHMRA, FOHTSF
          .EXTRN ERMS, FOOFIL, OUTLIN
          .EXTRN OUTPAS
          .PSECT $CODE$,NOWRT,2

```

```

          OFFC 0000
          .ENTRY FOOOUT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- ; 0239
          R11
          MOVAB FNCT+4, R11
          MOVL OUTPUT_COUNT, R0 ; 0280
          BGTR 1$
          RET
          6B      50 D1 00010 1$: CMPL R0, FNCT+4 ; 0286
          1A 15 00013 BLEQ 2$
          06 DD 00015 PUSHL #6 ; 0289

```

		00000000'	EF	9F	00017		PUSHAB	P.AAA			
		00000000G	8F	DD	0001D		PUSHL	#RNFILE			
	C0000000G		EF	03	FB	00023	CALLS	#3, ERMS			
			57	6B	DO	0002A	MOVL	FNCT+4, COUNT		0290	
				03	11	0002D	BRB	3\$		0286	
			57	50	DO	0002F	MOVL	R0, COUNT		0293	
				01	DD	00032	PUSHL	#1		0302	
	00000000G		EF	01	FB	00034	CALLS	#1, FOOFIL			
			01	50	D1	0003B	CMPL	R0, #1			
				19	13	0003E	BEQL	4\$			
				05	DD	00040	PUSHL	#5		0305	
	00000000G		EF	01	FB	00042	CALLS	#1, FOOFIL			
				06	DD	00049	PUSHL	#6		0306	
	00000000G		EF	01	FB	0004B	CALLS	#1, FOOFIL			
		FC		AB	7C	00052	CLRQ	FNCT		0307	
		08		AB	7C	00055	CLRQ	FNCT+12		0310	
				04	00058		RET			0304	
	14		AB	01	DO	00059	MOVL	#1, FNCT+24		0315	
			5A	00000000G	FF	DO	0005D	MOVL	@PHAN+40, HOLD_PAGING	0316	
				00000000G	FF	D4	00064	CLRL	@PHAN+40	0317	
	00000000G		EF	00000000G	EF	DO	0006A	MOVL	TSF, FOHTSF	0319	
	00000000G		EF	00000000G	EF	DO	00075	MOVL	MRA, FOHMRA	0320	
	00000000G		EF	00000000G	EF	9E	00080	MOVAB	FOOTSF, TSF	0321	
	00000000G		EF	00000000G	EF	9E	0008B	MOVAB	FOOMRA, MRA	0322	
				56	D4	00096	CLRL	FOO_FIRST_XTN		0323	
				53	D4	00098	CLRL	MAP_FIRST_XTN		0324	
			54	01	CE	0009A	MNEGL	#1, -I		0359	
				27	11	0009D	BRB	7\$			
	50		54	08	AB	C1	0009F	ADDL3	FNCT+12, I, R0	0360	
			58	00000000G	EF	40	DO	000A4	MOVL	FNISIZ[R0], R8	
						55	D4	000AC	CLRL	J	
					0080	31	000AE	BRW	11\$		
					08	DD	000B1	PUSHL	#8	0364	
	00000000G		EF	01	FB	000B3	CALLS	#1, FOOFIL			
			59	50	DO	000BA	MOVL	R0, STATUS			
				52	D4	000BD	CLRL	R2		0366	
			01	59	D1	000BF	CMPL	STATUS, #1			
				05	13	000C2	BEQL	8\$			
				52	D6	000C4	INCL	R2			
				00D1	31	000C6	BRW	17\$		0369	
			02	00000000G	EF	D1	000C9	CMPL	FOOREC, #2	0372	
					61	12	000D0	BNEQ	12\$		
					09	DD	000D2	PUSHL	#9	0377	
	00000000G		EF	01	FB	000D4	CALLS	#1, FOOFIL			
			59	50	DO	000DB	MOVL	R0, STATUS			
			01	59	D1	000DE	CMPL	STATUS, #1		0380	
				E3	12	000E1	BNEQ	7\$			
			51	00000000G	EF	DO	000E3	MOVL	FOOREC+12, R1	0386	
				2A	15	000EA	BLEQ	10\$			
			50	00000000G	FF	41	DE	000EC	MOVAL	@FOOREC+16[R1], R0	0394
				FC	A0	D5	000F4	TSTL	-4(R0)		
					1D	13	000F7	BEQL	10\$		
					56	D5	000F9	TSTL	FOO_FIRST_XTN	0398	
				0B	12	000FB	BNEQ	9\$			
			53	00000000G	EF	DO	000FD	MOVL	GCA+168, MAP FIRST_XTN	0403	
			56	FC	A0	DO	00104	MOVL	-4(R0), FOO_FIRST_XTN	0404	
	FC		A0	00000000	EF	DO	00108	MOVL	GCA+168, -4(R0)	0413	

			00000000G	EF	D6	00110		INCL	GCA+168		0414	
			00000000G	EF	DD	00116	10\$:	PUSHL	FOOREC+4		0420	
				51	DD	0011C		PUSHL	R1		0419	
			00000000G	EF	DD	0011E		PUSHL	FOOREC+16			
			00000000G	EF	DD	00124		PUSHL	FOOREC+8			
		00000000G	EF	04	FB	0012A		CALLS	#4, OUTPAS			
				61	11	00131	11\$:	BRB	16\$			
				03	DD	00133	12\$:	PUSHL	#3		0425	
		00000000G	EF	01	FB	00135		CALLS	#1, FOOFIL			
			5B	52	E8	0013C		BLBS	R2, 17\$		0428	
			50	00000000G	EF	D0	0013F	MOVL	TSF, R0		0463	
			51	38	A0	D0	00146	MOVL	56(R0), R1			
					3F	13	0014A	BEQL	15\$			
			26	14	A0	E9	0014C	BLBC	20(R0), 14\$		0467	
					56	D5	00150	TSTL	FOO_FIRST_XTN		0472	
					0A	12	00152	BNEQ	13\$			
			53	00000000G	EF	D0	00154	MOVL	GCA+168, MAP_FIRST_XTN		0477	
			56	51	D0	0015B		MOVL	R1, FOO_FIRST_XTN		0478	
		38	A0	00000000G	EF	D0	0C15E	13\$:	MOVL	GCA+168, 56(R0)	0487	
		3C	A0	00000000G	EF	D0	00166		MOVL	GCA+168, 60(R0)	0488	
					EF	D6	0016E		INCL	GCA+168	0489	
					15	11	00174		BRB	15\$	0467	
		52	51	56	C3	00176	14\$:	SUBL3	FOO_FIRST_XTN, R1, XTN_OFFSET		0508	
			52	53	C0	0017A		ADDL2	MAP_FIRST_XTN, R2		0514	
		3C	A0	51	C3	0017D		SUBL3	R1, -60(R0), R1			
			51	52	C1	00182		ADDL3	R2, R1, 60(R0)			
			38	52	D0	00187		MOVL	R2, 56(R0)		0517	
				7E	D4	0018B	15\$:	CLRL	-(SP)		0527	
			00000000G	EF	01	FB	0018D		CALLS	#1, OUTLIN		
		FF17	55	58	F1	00194	16\$:	ACBL	R8, #1, J, 6\$		0360	
			02	57	F2	0019A	17\$:	AOBLSS	COUNT, I, 18\$			
				03	11	0019E		BRB	19\$			
				FEFC	31	001A0	18\$:	BRW	5\$			
			00000000G	EF	D0	001A3	19\$:	MOVL	FOHTSF, TSF		0532	
			00000000G	EF	D0	001AE		MOVL	FOHMRA, MRA		0533	
			00000000G	FF	5A	D0	001B9		MOVL	HOLD_PAGING, @PHAN+40		0534
				6B	57	C2	001C0		SUBL2	COUNT, FNCT+4		0535
				FC	57	C2	001C3		SUBL2	COUNT, FNCT		0536
			08	AB	57	C0	001C7		ADDL2	COUNT, FNCT+12		0537
					14	AB	D4	001CB	CLRL	FNCT+24		0538
				FC	AB	D5	001CE		TSTL	FNCT		0541
					17	14	001D1		BGTR	20\$		
			00000000G	EF	05	DD	001D3		PUSHL	#5		0545
					01	FB	001D5		CALLS	#1, FOOFIL		
			00000000G	EF	06	DD	001DC		PUSHL	#6		0546
					01	FB	001DE		CALLS	#1, FOOFIL		
					6B	D4	001E5		CLRL	FNCT+4		0548
					08	AB	7C	001E7	CLRQ	FNCT+12		0547
					04	001EA	20\$:	RET			0552	

: Routine Size: 491 bytes, Routine Base: \$CODE\$ + 0000

: 426 0553 1
: 427 0554 1 END
: 428 0555 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	8	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODES	491	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.2
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	37	2	86	00:00.3

COMMAND QUALIFIERS

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

: Size: 491 code + 8 data bytes
 : Run Time: 00:10.3
 : Elapsed Time: 00:31.4
 : Lines/CPU Min: 3229
 : Lexemes/CPU-Min: 15660
 : Memory Used: 119 pages
 : Compilation Complete

