



```

FFFFFFFFF NN NN 000000 NN NN LL YY YY
FFFFFFFFF NN NN 000000 NN NN LL YY YY
FF NN NN 00 00 NN NN LL YY YY
FF NN NN 00 00 NN NN LL YY YY
FF NNNN NN 00 00 NNNN NN LL YY YY
FF NNNN NN 00 00 NNNN NN LL YY YY
FFFFFFFF NN NN NN 00 00 NN NN NN LL YY
FFFFFFFF NN NN NN 00 00 NN NN NN LL YY
FF NN NNNN 00 00 NN NNNN LL YY
FF NN NNNN 00 00 NN NNNN LL YY
FF NN NN 00 00 NN NN LL YY
FF NN NN 00 00 NN NN LL YY
FF NN NN 00 00 NN NN LL YY
FF NN NN 000000 NN NN LLLLLLLLLL YY
FF NN NN 000000 NN NN LLLLLLLLLL YY

```

```

LL I I I I I S S S S S S S
LL I I I I I S S S S S S S
LL I I S S
LL I I S S
LL I I S S
LL I I S S
LL I I S S S S S S
LL I I S S S S S S
LL I I S S
LL I I S S
LL I I S S
LL I I S S
LLLLLLLLLLL I I I I I S S S S S S S
LLLLLLLLLLL I I I I I S S S S S S S

```

```

1 0001 0 %TITLE 'Process .FN and .END FOOTNOTE directives'
2 0002 0 MODULE fnonly ( 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 ++
33 0033 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
34 0034 1
35 0035 1 ABSTRACT: Processes the .FOOTNOTE and .END FOOTNOTE commands.
36 0036 1
37 0037 1 ENVIRONMENT: Transportable
38 0038 1
39 0039 1 AUTHOR: R.W.Friday CREATION DATE: September, 1978
40 0040 1

```

S  
R  
E  
L  
C

```

: 42 0041 1 %SBTTL 'Revision History'
: 43 0042 1 | MODIFIED BY:
: 44 0043 1 |
: 45 0044 1 | 029 RER00029 Ron Randall 14-Apr-1983
: 46 0045 1 | Fixed bug in footnote numbering code.
: 47 0046 1 |
: 48 0047 1 | 028 RER00028 Ron Randall 06-Apr-1983
: 49 0048 1 | For DSRPLUS: Added footnote numbering code.
: 50 0049 1 |
: 51 0050 1 | 027 KFA00027 Ken Alden 16-Mar-1983
: 52 0051 1 | PUSH/POP_sca now visible to DSR.
: 53 0052 1 |
: 54 0053 1 | 026 RER00026 Ron Randall 07-Mar-1983
: 55 0054 1 | Global edit of all modules. Updated module names, idents,
: 56 0055 1 | copyright dates. Changed require files to BLISS library.
: 57 0056 1 | --
: 58 0057 1 |

```

```

60 0058 1 %SBTTL 'Module Level Declarations'
61 0059 1
62 0060 1 : INCLUDE FILES:
63 0061 1
64 0062 1 LIBRARY 'NXPORT:XPORT'; : XPORT Library
65 0063 1 REQUIRE 'REQ:RNODEF'; : RUNOFF variant definitions
66 0194 1
67 U 0195 1 %IF DSRPLUS %THEN
68 U 0196 1 LIBRARY 'REQ:DPLLIB'; : DSRPLUS BLISS Library
69 0197 1 %ELSE
70 0198 1 LIBRARY 'REQ:DSRLIB'; : DSR BLISS Library
71 0199 1 %FI
72 0200 1
73 0201 1
74 0202 1 : EXTERNAL REFERENCES:
75 0203 1
76 0204 1 EXTERNAL LITERAL
77 0205 1 RINTES : UNSIGNED (8);
78 0206 1
79 0207 1 EXTERNAL
80 0208 1 FNCT : FNCT_DEFINITION, : General information about footnotes
81 0209 1 FNESIZ : FN_EXT_SIZE_DEFINITION, : Lines in each pending footnote
82 0210 1 FNISIZ : FN_INT_SIZE_DEFINITION, : TSF/MRA sets per pending footnote
83 0211 1 GCA : GCA_DEFINITION,
84 0212 1 IRA : FIXED_STRING,
85 0213 1 KHAR,
86 0214 1 LSTCHR : REF VECTOR,
87 0215 1 LSTLCH : REF VECTOR,
88 0216 1 LSTRCH : REF VECTOR,
89 0217 1 LSTLDD : REF VECTOR,
90 0218 1 LSTCNT : REF COUNTED_LIST,
91 0219 1 LSTSKP : REF VECTOR,
92 0220 1 MRA : REF FIXED_STRING,
93 0221 1 NUMPRM : NUMPRM_DEFINE,
94 0222 1 PDT : REF PDT_DEFINITION,
95 0223 1 PHAN : PHAN_DEFINITION,
96 0224 1 SCA : SCA_DEFINITION,
97 0225 1 TSF : TSF_DEFINITION;
98 0226 1
99 0227 1 EXTERNAL
100 0228 1 FOOMRA : FIXED_STRING, :Substitute MRA.
101 0229 1 FOOPDT : VECTOR [PDT_SIZE], :Save area for paragraph stuff.
102 0230 1 FOOSCA : VECTOR [SCA_SIZE], :Save area for SCA
103 0231 1 F_LIST : COUNTED_LIST [3], :Substitute .LIST counters
104 0232 1 F_LCHR : VECTOR [3], :Substitute .LIST bullet char.
105 0233 1 F_LSKP : VECTOR [3], :Substitute .LIST EL spacing.
106 0234 1 F_LLCHR : VECTOR [3], :Substitute left character
107 0235 1 F_LRCHR : VECTOR [3], :Substitute right character
108 0236 1 F_LDD : VECTOR [3], :Substitute display descriptor.
109 0237 1 FDOTSF : VECTOR [TSF_SIZE]; :Substitute TSF area
110 0238 1
111 0239 1 EXTERNAL
112 0240 1 FOHLCH,
113 0241 1 FOHRCH,
114 0242 1 FOHDD,
115 0243 1 FOHLSL,
116 0244 1 FOHLSK,

```

```

: 117 0245 1 FOHLST,
: 118 0246 1 FOHPDT,
: 119 0247 1 FOHMRA,
: 120 0248 1 FOHTSF;
: 121 0249 1
: 122 0250 1 EXTERNAL LITERAL
: 123 0251 1 RNF GFC,
: 124 0252 1 RNFTMF,
: 125 0253 1 RNFUME;
: 126 0254 1
: 127 0255 1 EXTERNAL LITERAL
: 128 0256 1 S_FMRA;
: 129 0257 1
: 130 U 0258 1 %IF DSRPLUS %THEN
: 131 U 0259 1 EXTERNAL ROUTINE
: 132 U 0260 1 GUSKIP,
: 133 U 0261 1 OUTCRG;
: 134 0262 1 %FI
: 135 0263 1
: 136 0264 1 EXTERNAL ROUTINE
: 137 0265 1 ERM,
: 138 0266 1 ERMA,
: 139 0267 1 ERML,
: 140 0268 1 ERMN,
: 141 0269 1 FOOFIL,
: 142 0270 1 OUTNJ,
: 143 0271 1 SETCAS;
: 144 0272 1
: 145 0273 1 : OWN STORAGE:
: 146 0274 1 :
: 147 0275 1 OWN
: 148 0276 1 PP_SCA · $H_R_SCA_BLOCK; !Used in PUSH_SCA, POP_SCA macros (defined in SCA.REQ).
: 149 0277 1

```

!Allocated size for footnote MRA.

```

151 0278 1 %SBTTL 'FN -- body of routine'
152 0279 1 GLOBAL ROUTINE fn (handler_code) : NOVALUE =
153 0280 1
154 0281 1 |++
155 0282 1 | FUNCTIONAL DESCRIPTION:
156 0283 1 |
157 0284 1 |     Processes the .FOOTNOTE and .END FOOTNOTE commands.
158 0285 1 |
159 0286 1 | FORMAL PARAMETERS:
160 0287 1 |
161 0288 1 |     handler_code - Indicates which command is to be processed.
162 0289 1 |
163 0290 1 | IMPLICIT INPUTS:
164 0291 1 |
165 0292 1 |     numprm - Contains a number, as processed by GETNUM.
166 0293 1 |
167 0294 1 | IMPLICIT OUTPUTS:      None
168 0295 1 |
169 0296 1 | ROUTINE VALUE:
170 0297 1 | COMPLETION CODES:      None
171 0298 1 |
172 0299 1 | SIDE EFFECTS:          None
173 0300 1 | --
174 0301 1 |
175 0302 2 |     BEGIN
176 0303 2 |     LOCAL                                !To interchange GCA_NORMAL_XTN and
177 0304 2 |         HOLD_FOOT_XTN;                  !GCA_FOOT_XTN between .fn and .efn.
178 0305 2 |
179 0306 2 |     SELECTONE .HANDLER_CODE OF
180 0307 2 |         SET
181 0308 2 |
182 0309 2 |         [H_FOOTNOTE] :
183 0310 2 |             BEGIN
184 0311 2 |
185 U 0312 2 | %IF DSRPLUS %THEN
186 UU 0313 2 |     |
187 UU 0314 2 |     |     If numbering footnotes, set up variables.
188 UU 0315 2 |     |
189 UU 0316 2 |     |     IF .FNCT_NUMBERING
190 UU 0317 2 |     |     THEN
191 UU 0318 2 |     |         BEGIN
192 UU 0319 2 |     |         |
193 UU 0320 2 |     |         |     Flag the start of a numbered footnote.
194 UU 0321 2 |     |         |
195 UU 0322 2 |     |         |     FNCT_FIRST_LINE = 1;
196 UU 0323 2 |     |         |
197 UU 0324 2 |     |         |     Bump footnote number by one.  Get left and right digits right.
198 UU 0325 2 |     |         |
199 UU 0326 2 |     |         |     FNCT_NUMBER_R = .FNCT_NUMBER_R + 1;
200 UU 0327 2 |     |         |     FNCT_NUMBER_L = .FNCT_NUMBER_L + 1;
201 UU 0328 2 |     |         |
202 UU 0329 2 |     |         |     IF .FNCT_NUMBER_R EQL 10
203 UU 0330 2 |     |         |     THEN
204 UU 0331 2 |     |         |         BEGIN
205 UU 0332 2 |     |         |         |
206 UU 0333 2 |     |         |         |     FNCT_NUMBER_R = 0;
207 U 0334 2 |     |         |         |     FNCT_NUMBER_L = .FNCT_NUMBER_L + 1;
207 U 0334 2 |     |         |         |     END;

```

```

208 U 0335 3
209 U 0336 3
210 0337 XFI
211 0338
212 0339
213 0340 GCA_CONCAT = FALSE; !Don't allow .NO SPACE
214 0341 FNCT_N = .NUM_VALUE; !Save specified count.
215 0342
216 0343 IF FOOFIL (FOO_OPOU) NEQ FOO_NORMAL
217 0344 THEN
218 0345 !Couldn't open footnote file.
219 0346 RETURN;
220 0347 !Check for too many footnotes.
221 0348 IF .FNCT_COUNT GEQ FNCT_MAX
222 0349 THEN
223 0350 !Tell the user that he's attempting to define too many footnotes.
224 0351 !Note that everything proceeds as if nothing was wrong.
225 0352 !However, what eventually happens, when .END FOOTNOTE occurs,
226 0353 !is that the footnotes get merged together. As soon as a footnote
227 0354 !gets expanded and output however, everything continues along ok.
228 0355 !Note that relevant data structures have extra space allocated so
229 0356 !special casing can be pretty much avoided in this situation.
230 0357 ERML (RNFTMF);
231 0358
232 0359 FNCT_COLLECTING = TRUE; !Officially collecting footnotes now.
233 0360 FNESTZ [.FNCT_COUNT] = 0; !Clear external size.
234 0361 FNISIZ [.FNCT_COUNT] = 0; !Clear internal size.
235 0362 FOHTSF = .TSF; !Save previous status of SCANT.
236 0363 FOHMRA = .MRA;
237 0364 !Save entire status of SCA.
238 0365
239 0366 PUSH_SCA; !Save the special SCA bits that are SAVED.
240 0367
241 0368 INCR I FROM 0 TO SCA_SIZE - 1 DO
242 0369 FOOSCA [.I] = .SCA [.I];
243 0370
244 0371 ! Initialize SCA to footnote environment.
245 0372 SCA_LM = 0;
246 0373 SCA_FC = TRUE;
247 0374 SCA_FC_CASE = TRUE;
248 0375 SCA_NBITS = FALSE;
249 0376 SCA_WRD_NBITS = FALSE;
250 0377 SCA_WRD_CNBITS = FALSE;
251 0378 SCA_WRD_ACNBITS = FALSE;
252 0379 SCA_WRD_FOOTW = 0;
253 0380 SCA_SECT_EMPTY = FALSE;
254 0381 SCA_FILL = TRUE;
255 0382 SCA_JUSTIFY = .GCA AUTOJUST OR .SCA_JUSTIFY; ! Obey .[no]autojustify directive
256 0383 SCA_CROCK = .SCA_JUSTIFY; ! Obey .[no]autojustify directive
257 0384 SCA_WRD_LST_UND = 0;
258 0385 SCA_WRD_LST_SP = 0;
259 0386 SCA_WRD_LST_JUS = 0;
260 0387 SCA_WRD_INT_L = 0;
261 0388 SCA_WRD_EXT_L = 0;
262 0389 SCA_WRD_F_XTN = 0;
263 0390 SCA_WRD_L_XTN = 0;
264 0391 SCA_WRD_LT_PNCT = FALSE;

```



```

322 0449 3 FS INIT (MRA);
323 0450 SCA_WRD_PNTR = .FS NEXT (MRA);
324 0451 SCA_WRD_CPEND = RINTES;           !Current word is empty.
325 0452 HOLD FOOT_XTN = .GCA_FOOT_XTN; !Switch normal and footnote
326 0453 GCA_FOOT_XTN = .GCA_NORMAL_XTN; !transaction numbers, to remove
327 0454 GCA_NORMAL_XTN = .HOLD_FOOT_XTN; !FNCT dependency in OUTLIN.
328 0455 END;
329 0456
330 0457 [H_END FOOTNOTE] :
331 0458 BEGIN
332 0459
333 0460 IF NOT .FNCT_COLLECTING
334 0461 THEN
335 0462 !User said .END FOOT without first saying .FN.
336 0463 BEGIN
337 0464 ERMA (RNFUME, FALSE);
338 0465 RETURN;
339 0466 END;
340 0467
341 0468 OUTNJ ();           !Terminate pending text.
342 0469 !NOTE:*** It is important that the call on OUTNJ *NOT* be removed.
343 0470 !This is because the routine PUS, when it detects an ENDFOOTNOTE flag,
344 0471 !has no way of telling this routine via DOCM to call OUTNJ. If there
345 0472 !were no ENDFOOTNOTE flag, or if the user says .END FOOTNOTE, there is
346 0473 !no problem, since the entries in RUNTAB.REQ for .END FOOTNOTE do
347 0474 !indicate OUTNJ is to be called.
348 0475
349 U 0476 %IF DSRPLUS %THEN
350 UU 0477
351 UU 0478     Put out a blank line separator for numbered footnotes.
352 UU 0479
353 UU 0480 IF .FNCT_NUMBERING
354 UU 0481 THEN
355 UU 0482 BEGIN
356 UU 0483 GUSKIP (1);
357 UU 0484 OUTCRG ();
358 UU 0485
359 UU 0486     Set flag to indicate the end of a footnote.
360 UU 0487     (This flag is normally set to 0. At the very start of
361 UU 0488     a footnote, it is set to 1 to indicate the first line.
362 UU 0489     At the end it is set to 2 to indicate the end. It is
363 UU 0490     tested and reset to 0 (in FOOUM) very quickly.)
364 UU 0491
365 UU 0492 FNCT_FIRST_LINE = 2;
366 U 0493 END;
367 0494 %FI
368 0495
369 0496 FNCT_COLLECTING = FALSE;           !Officially not collecting footnotes now.
370 0497
371 0498 INCR I FROM 0 TO SCA_SIZE - 1 DO
372 0499     SCA [I] = .FOOSCA [I];       !Restore previous SCA.
373 0500
374 0501 POP_SCA;       !Restore the special SCA bits
375 0502
376 0503 !If the footnote tables have not overflowed, update the count of footnotes
377 0504 !associated with this word.
378 0505 IF (.FNCT_COUNT LSS FNCT_MAX) AND

```

```
379 0506 3 .PHAN_PAGING !Merge all footnotes if .NO PAGING
380 0507 THEN
381 0508 SCA_WRD_FOOTW = .SCA_WRD_FOOTW + 1;
382 0509
383 0510 MRA = .FOHMRA; !Restore previous MRA.
384 0511 TSF = .FOHTSF; !Restore previous TSF.
385 0512 PDT = .FOHPDT; !Restore previous paragraph settings.
386 0513 LSTCNT = .FOHLST; !Restore previous .LIST information.
387 0514 LSTCHR = .FOHLSC;
388 0515 LSTSKP = .FOHLSK;
389 0516 LSTLCH = .FOHLCH;
390 0517 LSTRCH = .FOHRCH;
391 0518 LSTLDD = .FOHDD;
392 0519
393 0520 !If the user used the outmoded .FN n command (i.e., specified what he believed
394 0521 !the number of footnote lines to be) verify that he indeed gave a correct
395 0522 !value for 'n'.
396 0523 IF (.FNCT_N NEQ .FNESIZ [.FNCT_COUNT]) AND
397 0524 (.FNCT_N NEQ 0)
398 0525 THEN
399 0526 !User specified a count and it doesn't agree with how big the footnote really is.
400 0527 !Tell the user he made a mistake, and also how big the footnote really is.
401 0528 ERMN (RNFGFC, .FNESIZ [.FNCT_COUNT]);
402 0529
403 0530 !Update some footnote statistics.
404 0531 !Note that three situations occur here, one of which is an error condition.
405 0532 IF .FNCT_COUNT GEQ FNCT_MAX
406 0533 THEN
407 0534 !The error case. Statistics have been piling up in an extra
408 0535 !location in the tables. Merge that info with other information
409 0536 !so that the footnotes get merged.
410 0537 BEGIN
411 0538 FNISIZ [FNCT_MAX - 1] = .FNISIZ [.FNCT_COUNT - 1] + .FNISIZ [.FNCT_COUNT];
412 0539 FNESIZ [FNCT_MAX - 1] = .FNESIZ [.FNCT_COUNT - 1] + .FNESIZ [.FNCT_COUNT]
413 0540 END
414 0541 ELSE
415 0542
416 0543 !The normal situation. The tables won't overflow, and so all we have to
417 0544 !do is update footnote count information.
418 0545 IF .PHAN_PAGING
419 0546 THEN
420 0547 !The document is being paged, so don't merge footnotes.
421 0548 BEGIN
422 0549 FNCT_COUNT = .FNCT_COUNT + 1;
423 0550 FNCT_WAITING = .FNCT_WAITING + 1;
424 0551 END
425 0552 ELSE
426 0553
427 0554 !The document is not being paged.
428 0555 !Merge all footnotes into one biggie.
429 0556 IF .FNCT_COUNT EQL 0
430 0557 THEN
431 0558 !This is the first footnote in the .NOPAGING environment
432 0559 BEGIN
433 0560 FNCT_COUNT = 1;
434 0561 FNCT_READY = 1;
435 0562 END
```

```

: 436 0563 3 ELSE
: 437 0564 3 !Merge new footnote with first footnote
: 438 0565 4 BEGIN
: 439 0566 4 !Note that FNCT_COUNT will not necessarily be 1. That's because
: 440 0567 4 !the user may have entered no paging mode after having accumulated
: 441 0568 4 !some footnotes in paging mode. In any case, this new footnote gets
: 442 0569 4 !arbitrarily counted in with the size of the very first footnote.
: 443 0570 4 FNESIZ [0] = .FNESIZ [0] + .FNESIZ [.FNCT_COUNT];
: 444 0571 4 FNISIZ [0] = .FNISIZ [0] + .FNISIZ [.FNCT_COUNT];
: 445 0572 3 END;
: 446 0573 3
: 447 0574 3 HOLD FOOT_XTN = .GCA_FOOT_XTN; !Switch normal and footnote
: 448 0575 3 GCA_FOOT_XTN = .GCA_NORMAL_XTN; !transaction numbers, to remove
: 449 0576 3 GCA_NORMAL_XTN = .HOLD_FOOT_XTN; !FNCT dependency in OUTLIN.
: 450 0577 2 END;
: 451 0578 2
: 452 0579 2 TES;
: 453 0580 2
: 454 0581 1 END;

```

!End of FN

.TITLE FNONLY Process .FN and .END FOOTNOTE directives  
.IDENT \V04-000\

.PSECT \$OWNS,NOEXE,2

00000 PP\_SCA: .BLKB 48

.EXTRN RINTES, FNCT, FNESIZ  
.EXTRN FNISIZ, GCA, IRA  
.EXTRN KHAR, LSTCHR, LSTLCH  
.EXTRN LSTRCH, LSTLDD, LSTCNT  
.EXTRN LSTSKP, MRA, NUMPRM  
.EXTRN PDT, PHAN, SCA, TSF  
.EXTRN FOOMRA, FOOPDT, FOOSCA  
.EXTRN F\_LIST, F\_LCHR, F\_LSKP  
.EXTRN F\_LLCHR, F\_LRCHR  
.EXTRN F\_LDD, FOOTSF, FOHLCH  
.EXTRN FOHRCH, FOHDD, FOHLSC  
.EXTRN FOHLSK, FOHLST, FOHPDT  
.EXTRN FOHMRA, FOHTSF, RNFGFC  
.EXTRN RNFTMF, RNFUME, S\_FMRA  
.EXTRN ERM, ERMA, ERML  
.EXTRN ERMN, FOFIL, OUTNJ  
.EXTRN SETCAS

.PSECT \$CODE\$,NOWRT,2

OFFC 00000  
5B 00000000G EF 9E 00002  
5A 00000000G EF 9E 00009  
59 00000000G EF 9E 00010  
58 00000000G EF 9E 00017  
57 00000000G EF 9E 0001E  
56 00000000G EF 9E 00025  
55 00000000G EF 9E 0002C  
54 00000000G EF 9E 00033

.ENTRY FN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11  
MOVAB LSTCNT, R11  
MOVAB PDT, R10  
MOVAB MRA, R9  
MOVAB TSF, R8  
MOVAB FNISIZ, R7  
MOVAB GCA+172, R6  
MOVAB FNESIZ, R5  
MOVAB FNCT, R4

: 0279 |

	53	00000000'	EF	9E	0003A	MOVAB	PP_SCA, R3		
	52	00000000G	EF	9E	00041	MOVAB	SCA+100, R2		
0000005A	50	04	AC	D0	00048	MOVL	HANDLER_CODE, R0		0306
	8F		50	D1	0004C	CML	R0, #90		0309
			03	13	00053	BEQL	1\$		
			J1D9	31	00055	BRW	7\$		
		84	A6	D4	00058	CLRL	GCA+48		0339
08	A4	00000000G	EF	D0	0005B	MOVL	NUMPRM+4, FNCT+8		0340
			02	DD	00063	PUSHL	#2		0342
00000000G	EF		01	FB	00065	CALLS	#1, FOOFIL		
	01		50	D1	0006C	CML	R0, #1		
			01	13	0006F	BEQL	2\$		
				04	00071	RET			
	14		64	D1	00072	CML	FNCT, #20		0348
			0D	19	00075	BLSS	3\$		
00000000G		00000000G	8F	DD	00077	PUSHL	#RNFTMF		0357
	EF		01	FB	0007D	CALLS	#1, ERML		
14	A4		01	D0	00084	MOVL	#1, FNCT+20		0359
	50		64	D0	00088	MOVL	FNCT, R0		0360
			6540	D4	0008B	CLRL	FNESIZ[R0]		
			6740	D4	0008E	CLRL	FNISIZ[R0]		0361
00000000G	EF		68	D0	00091	MOVL	TSF, FOHTSF		0362
00000000G	EF		69	D0	00098	MOVL	MRA, FOHMRA		0363
	63	00	B2	D0	0009F	MOVL	@SCA+100, PP_SCA		
04	A3	04	B2	D0	000A3	MOVL	@SCA+104, PP_SCA+4		
08	A3	08	B2	D0	000A8	MOVL	@SCA+108, PP_SCA+8		
0C	A3	0C	B2	D0	000AD	MOVL	@SCA+112, PP_SCA+12		
10	A3	10	B2	D0	000B2	MOVL	@SCA+116, PP_SCA+16		
14	A3	14	B2	D0	000B7	MOVL	@SCA+120, PP_SCA+20		
18	A3	18	B2	D0	000BC	MOVL	@SCA+124, PP_SCA+24		
1C	A3	1C	B2	D0	000C1	MOVL	@SCA+128, PP_SCA+28		
20	A3	20	B2	D0	000C6	MOVL	@SCA+132, PP_SCA+32		
24	A3	24	B2	D0	000CB	MOVL	@SCA+136, PP_SCA+36		
28	A3	28	B2	D0	000D0	MOVL	@SCA+140, PP_SCA+40		
2C	A3	2C	B2	D0	000D5	MOVL	@SCA+144, PP_SCA+44		
			50	D4	000DA	CLRL	I		0368
00000000GEF	40	9C	A240	D0	000DC	MOVL	SCA[I], FOOSCA[I]		0369
EE	50	0000005F	8F	F3	000E6	AOBLEQ	#95, I, 4\$		
			10	B2	D4	000EE	CLRL	@SCA+116	0372
	6C	A2	01	D0	000F1	MOVL	#1, SCA+208		0374
	30	A2	01	7D	000F5	MOVQ	#1, SCA+148		0373
			5C	A2	7C	000F9	CLRQ	SCA+192	0376
			64	A2	D4	000FC	CLRL	SCA+200	0378
			50	A2	D4	000FF	CLRL	SCA+180	0380
	04	B2	01	D0	00102	MOVL	#1, @SCA+104		0381
	00	B2	FF64	D6	C8	00106	BISL2	@GCA+16, @SCA+100	0382
	0C	B2	00	B2	D0	0010C	MOVL	@SCA+100, @SCA+112	0383
			00E8	C2	D4	00111	CLRL	SCA+332	0385
			00EC	C2	7C	00115	CLRQ	SCA+336	0386
			0098	C2	7C	00119	CLRQ	SCA+252	0387
			00C0	C2	7C	0C11D	CLRQ	SCA+292	0379
			00C8	C2	D4	00121	CLRL	SCA+300	0390
			00DC	C2	D4	00125	CLRL	SCA+320	0392
			00E0	C2	7C	00129	CLRQ	SCA+324	0393
			D4	B6	DD	0012D	PUSHL	@GCA+128	0396
00000000G	EF		01	FB	00130	CALLS	#1, SETCAS		
			50	D4	00137	CLRL	I		0401

F2	00000000G	EF	40	00	BA	D0	00139	5\$:	MOVL	@PDT[1], FOOPDT[1]	:	
	50				02	F3	00143		AOBLEQ	#2, 1, 5\$	:	0403
	00000000G	EF			6A	D0	00147		MOVL	PDT, FOHPDT	:	0404
				00000000G	EF	9E	0014E		MOVAB	FOOPDT, PDT	:	0407
	00000000G	EF			6B	D0	00155		MOVL	LSTCNT, FOHLST	:	0408
	00000000G	EF		00000000G	EF	D0	0015C		MOVL	LSTCHR, FOHLSC	:	0409
	C0000000G	EF		00000000G	EF	D0	00167		MOVL	LSTSKP, FOHLSC	:	0410
	00000000G	EF		00000000G	EF	D0	00172		MOVL	LSTLCH, FOHLCH	:	0411
	00000000G	EF		00000000G	EF	D0	0017D		MOVL	LSTRCH, FOHRCH	:	0412
	00000000G	EF		00000000G	EF	D0	00188		MOVL	LSTLDD, FOHDD	:	0413
				00000000G	EF	9E	00193		MOVAB	F_LIST, LSTCNT	:	0414
	00000000G	EF		00000000G	EF	9E	0019A		MOVAB	F_LCHR, LSTCHR	:	0415
	00000000G	EF		00000000G	EF	9E	001A5		MOVAB	F_LSKP, LSTSKP	:	0416
	00000000G	EF		00000000G	EF	9E	001B0		MOVAB	F_LLCHR, LSTLCH	:	0417
	00000000G	EF		00000000G	EF	9E	001BB		MOVAB	F_LRCHR, LSTRCH	:	0418
	00000000G	EF		00000000G	EF	9E	001C6		MOVAB	F_LDD, LSTLDD	:	0419
						6B	001D1		MOVL	LSTCNT, R0	:	
						03	001D4		MOVL	#3, (R0)	:	
	04					01	7D 001D7		MOVQ	#1, 4(R0)	:	0420
				00000000G	FF	D4	001DB		CLRL	@LSTCHR	:	0422
				00000000G	FF	D4	001E1		CLRL	@LSTLCH	:	0423
	00000000G	FF			2E	D0	001E7		MOVL	#46, @LSTRCH	:	0424
				00000000G	FF	D4	001EE		CLRL	@LSTLDD	:	0425
				00000000G	EF	9E	001F4		MOVAB	FOOTSF, TSF	:	0426
				00000000G	EF	9E	001FB		MOVAB	FOOMRA, MRA	:	0427
						50	D4 00202		CLRL	I	:	0430
F8						00	B840	6\$:	CLRL	@TSF[1]	:	
						27	F3 00208		AOBLEQ	#39, 1, 6\$	:	
						69	D0 0020C		MOVL	MRA, R0	:	0445
	08			00000000G	8F	D0	0020F		MOVL	#5, FMRA, 8(R0)	:	
						0C	A0 D4 00217		CLRL	12(R0)	:	0449
						10	A0 9E 0021A		MOVAB	16(R0), (R0)	:	
	04					60	D0 0021E		MOVL	(R0), 4(R0)	:	
	0094			04		A0	D0 00222		MOVL	4(R0), SCA+248	:	0450
	00B4			00G		8F	9A 00228		MOVZBL	#RINTES, SCA+280	:	0451
						0132	31 0022E		BRW	16\$	:	0452
						50	D1 00231	7\$:	CPL	R0, #57	:	0457
						01	13 00234		BEQL	8\$	:	
						04	00236		RET		:	
						14	A4 E8 00237	8\$:	BLBS	FNCT+20, 9\$	:	0460
						7E	D4 0023B		CLRL	-(SP)	:	0464
				00000000G	8F	DD	0023D		PUSHL	#RNFUME	:	
	00000000G	EF			02	FB	00243		CALLS	#2, ERMA	:	
						04	0024A		RET		:	0463
	00000000G	EF				00	FB 0024B	9\$:	CALLS	#0, OUTNJ	:	0468
						14	A4 D4 00252		CLRL	FNCT+20	:	0496
						50	D4 00255		CLRL	I	:	0498
EE	9C	A240	00000000G	EF	40	D0	00257	10\$:	MOVL	FOOSCA[1], SCA[1]	:	0499
			0000005F		8F	F3	00261		AOBLEQ	#95, 1, 10\$	:	
	00	B2			63	D0	00269		MOVL	PP_SCA, @SCA+100	:	
	04	B2		04	A3	D0	0026D		MOVL	PP_SCA+4, @SCA+104	:	
	08	B2		08	A3	D0	00272		MOVL	PP_SCA+8, @SCA+108	:	
	0C	B2		0C	A3	D0	00277		MOVL	PP_SCA+12, @SCA+112	:	
	10	B2		10	A3	D0	0027C		MOVL	PP_SCA+16, @SCA+116	:	
	14	B2		14	A3	D0	00281		MOVL	PP_SCA+20, @SCA+120	:	
	18	B2		18	A3	D0	00286		MOVL	PP_SCA+24, @SCA+124	:	
	1C	B2		1C	A3	D0	0028B		MOVL	PP_SCA+28, @SCA+128	:	

20	B2	20	A3	D0	00290	MOVL	PP_SCA+32, @SCA+132	0505		
24	B2	24	A3	D0	00295	MLVL	PP_SCA+36, @SCA+136			
28	B2	28	A3	D0	0029A	MOVL	PP_SCA+40, @SCA+140			
2C	B2	2C	A3	D0	0029F	MOVL	PP_SCA+44, @SCA+144			
	50		64	D0	002A4	MOVL	FNCT, R0			
	14		50	D1	002A7	CMPL	R0, #20			
			0B	18	002AA	BGEQ	11\$			
	04	00000000G	FF	E9	002AC	BLBC	@PHAN+40, 11\$	0506		
		00C0	C2	D6	002B3	INCL	SCA+292	0508		
	69	00000000G	EF	D0	002B7	11\$:	MOVL	FOHMRA, MRA	0510	
	68	00000000G	EF	D0	002BE	MOVL	FOHTSF, TSF	0511		
	6A	00000000G	EF	D0	002C5	MOVL	FOHPDT, PDT	0512		
	6B	00000000G	EF	D0	002CC	MOVL	FOHLST, LSTCNT	0513		
00000000G	EF	00000000G	EF	D0	002D3	MOVL	FOHLSC, LSTCHR	0514		
00000000G	EF	00000000G	EF	D0	002DE	MOVL	FOHLSK, LSTSKP	0515		
00000000G	EF	00000000G	EF	D0	002E9	MOVL	FOHLCH, LSTLCH	0516		
00000000G	EF	00000000G	EF	D0	002F4	MOVL	FOHRCH, LSTRCH	0517		
00000000G	EF	00000000G	EF	D0	002FF	MOVL	FOHDD, LSTLDD	0518		
	6540		08	A4	D1	0030A	CMPL	FNCT+8, FNESIZ[R0]	0523	
			15	13	0030F	BEQL	12\$			
			08	A4	D5	00311	TSTL	FNCT+8	0524	
			10	13	00314	BEQL	12\$			
			6540	DD	00316	PUSHL	FNESIZ[R0]	0528		
		00000000G	8F	DD	00319	PUSHL	#RNF GFC			
00000000G	EF		02	FB	0031F	CALLS	#2, ERMN			
	50		64	D0	00326	12\$:	MOVL	FNCT, R0	0532	
	14		50	D1	00329	CMPL	R0, #20			
			12	19	0032C	BLSS	13\$			
4C	A7	FC	A740	6740	C1	0032E	ADDL3	FNISIZ[R0], FNISIZ-4[R0], FNISIZ+76	0538	
4C	A5	FC	A540	6540	C1	00336	ADDL3	FNESIZ[R0], FNESIZ-4[R0], FNESIZ+76	0539	
				23	11	0033E	BRB	16\$		
		07	00000000G	FF	E9	00340	13\$:	BLBC	@PHAN+40, 14\$	0545
				64	D6	00347	INCL	FNCT	0549	
			10	A4	D6	00349	INCL	FNCT+16	0550	
				15	11	0034C	BRB	16\$	0545	
				50	D5	0034E	14\$:	TSTL	R0	0556
				09	12	00350	BNEQ	15\$		
	64			01	D0	00352	MOVL	#1, FNCT	0560	
04	A4			01	D0	00355	MOVL	#1, FNCT+4	0561	
				08	11	00359	BRB	16\$	0556	
	65		6540	C0	0035B	15\$:	ADDL2	FNESIZ[R0], FNESIZ	0570	
	67		6740	C0	0035F	ADDL2	FNISIZ[R0], FNISIZ	0571		
	50		66	D0	00363	16\$:	MOVL	GCA+172, HOLD_FOOT_XTN	0574	
	66	FC	A6	D0	00366	MOVL	GCA+168, GCA+T72	0575		
	FC	A6	50	D0	0036A	MOVL	HOLD_FOOT_XTN, GCA+168	0576		
				04	0036E	RET		0581		

; Routine Size: 879 bytes, Routine Base: \$CODE\$ + 0000

: 455 0582 1  
: 456 0583 1 END  
: 457 0584 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	48	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	879	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	00:00.1
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	56	7	86	00:00.3

COMMAND QUALIFIERS

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

: Size: 879 code + 48 data bytes  
 : Run Time: 00:24.1  
 : Elapsed Time: 00:52.1  
 : Lines/CPU Min: 1455  
 : Lexemes/CPU-Min: 15198  
 : Memory Used: 189 pages  
 : Compilation Complete

ENDWRD LIS	ERROR LIS	FIGURE LIS	FLGSEM LIS	FOOFIL LIS	GCODE LIS
FCTMRA LIS	FENONLY LIS	FJFNFI LIS	FOOBOT LIS	GBLDCL LIS	
FNDPLG LIS	FOOOUT LIS	FORMAT LIS			