



```

FFFFFFFFF 000000 RRRRRRRR UU UU DDDDDDDD FFFFFFFFFF WW WW NN NN
FFFFFFFFF 000000 RRRRRRRR UU UU DDDDDDDD FFFFFFFFFF WW WW NN NN
FF 00 00 RR RR UU UU DD DD FF WW WW NN NN
FF 00 00 RR RR UU UU DD DD FF WW WW NN NN
FF 00 00 RR RR UU UU DD DD FF WW WW NNNN NN
FFFFFFFF 00 00 RRRRRRRR UU UU DD DD FFFFFFFF WW WW NN NN
FFFFFFFF 00 00 RRRRRRRR UU UU DD DD FFFFFFFF WW WW NN NN
FF 00 00 RR RR UU UU DD DD FF WW WW NN NN
FF 00 00 RR RR UU UU DD DD FF WW WW NN NN
FF 00 00 RR RR UU UU DD DD FF WWW WW NN NN
FF 00 00 RR RR UUUUUUUUU DDDDDDDD FF WWW WWW NN NN
FF 000000 RR RR UUUUUUUUU DDDDDDDD FF WW WW NN NN
FF 000000 RR RR UUUUUUUUU DDDDDDDD FF WW WW NN NN

```

```

LL IIIII SSSSSSS
LL IIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSS
LL II SSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLL IIIII SSSSSSS
LLLLLLLLL IIIII SSSSSSS

```

```

1 0001 0 MODULE FOR$$UDF_WN (%TITLE 'FOR$$UDF_WN - FORTRAN WRITE NAMELIST UDF Level'
2 0002 0      IDENT = '1-005'      . File: FORUDFWN.B32 Edit: SBL1005
3 0003 0      ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 *  ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 *  TRANSFERRED.
18 0018 1 *
19 0019 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 *  CORPORATION.
22 0022 1 *
23 0023 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY:      FORTRAN Compiled Code Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1      This module contains the User Data Formatter routines to perform
36 0036 1      FORTRAN NAMELIST WRITE statements.
37 0037 1
38 0038 1 ENVIRONMENT:  Runs at any access mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 29-August-1980
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 29-August-1980
45 0045 1 1-002 - Reflect group block spec change where count-of-variables is a word;
46 0046 1 second word is reserved. SBL 5-Dec-1980
47 0047 1 1-003 - Add text describing NAMELIST descriptor block. SBL 15-April-1981
48 0048 1 1-004 - REQUIRE FORERR.B32 instead of external reference to FOR$K_ symbols.
49 0049 1 SBL 12-Aug-1981
50 0050 1 1-005 - Add ability to dump just the variable names for "?" feature.
51 0051 1 Use prologue file. SBL 24-May-1983
52 0052 1 --
53 0053 1

```

```
55 0054 1 %SBTTL 'Declarations'
56 0055 1
57 0056 1 : PROLOGUE FILE:
58 0057 1 :
59 0058 1
60 0059 1 REQUIRE 'RTLIN:FORPROLOG';           ! FOR$ definitions
61 0125 1
62 0126 1 :
63 0127 1 : TABLE OF CONTENTS:
64 0128 1 :
65 0129 1
66 0130 1 FORWARD ROUTINE
67 0131 1   FOR$$UDF_WN0: JSB_UDF0 NOVALUE,       ! Start WRITE NAMELIST
68 0132 1   FOR$$DO_RML_OUTPUT: CALL_CCB NOVALUE, ! Do bulk of processing
69 0133 1   FOR$$UDF_WN9: JSB_UDF9 NOVALUE,       ! End WRITE NAMELIST
70 0134 1   CHECK_FIELD: CALL_CCB;               ! Check field width
71 0135 1
72 0136 1 :
73 0137 1 : MACROS:
74 0138 1 :
75 0139 1 :     NONE
76 0140 1 :
77 0141 1 : EQUATED SYMBOLS:
78 0142 1 :
79 0143 1 :     NONE
80 0144 1 :
81 0145 1 : FIELDS:
82 0146 1 :
83 0147 1 :     NONE
84 0148 1 :
85 0149 1 : OWN STORAGE:
86 0150 1 :
87 0151 1 :     NONE
88 0152 1 :
89 0153 1 : EXTERNAL REFERENCES:
90 0154 1 :
91 0155 1 :
92 0156 1 EXTERNAL ROUTINE
93 0157 1   FOR$$REC_WSNO: JSB_REC0 NOVALUE,       ! Set up for a write
94 0158 1   FOR$$REC_WSN1: JSB_REC1 NOVALUE,       ! Write a record
95 0159 1   FOR$$UDF_WL1: CALL_CCB NOVALUE,       ! Convert and move to buffer
96 0160 1   FOR$$SIGNAL_STO: NOVALUE;             ! Signal fatal error
```



FOR\$\$UDF\_WN  
1-005

FOR\$\$UDF\_WN - FORTRAN WRITE NAMELIST UDF Level  
FOR\$\$UDF\_WNO - Start WRITE NAMELIST

1 16  
16-Sep-1984 00:53:55  
14-Sep-1984 12:32:56

VAX-11 Bliss-32 V4.0-742  
[FORRTL.SRC]FORUDFWN.B32;1

Page 4  
(3)

```
                                .PSECT _FOR$CODE,NOWRT, SHR, PIC,2
                                00000000G 0G 16 0000 FOR$$UDF_WNO::
                                JSB     FOR$$REC_WSNO                : 0203
                                CLRL    -(SP)                        : 0205
0000V CF                        7E D4 00006                        :
                                01 FB 00008                        :
                                05 0000D                        : 0209
                                RSB
```

; Routine Size: 14 bytes, Routine Base: \_FOR\$CODE + 0000

```

: 148 0210 1 %SBTTL 'FOR$$DO NML OUTPUT - Do WRITE NAMELIST'
: 149 0211 1 GLOBAL ROUTINE FOR$$DO_NML_OUTPUT (
: 150 0212 1     NAMES_ONLY                               ! Set if only names are to be printed
: 151 0213 1     ): CALL_CCB NOVALUE =
: 152 0214 1
: 153 0215 1 !++
: 154 0216 1 ! FUNCTIONAL DESCRIPTION:
: 155 0217 1
: 156 0218 1     This routine performs one WRITE NAMELIST statement.
: 157 0219 1
: 158 0220 1 ! CALLING SEQUENCE:
: 159 0221 1
: 160 0222 1     CALL FOR$$DO_NML_OUTPUT (NAMES_ONLY.r.l.v)
: 161 0223 1
: 162 0224 1 ! FORMAL PARAMETERS:
: 163 0225 1
: 164 0226 1     NAMES_ONLY                               0 if both names and values are to be printed
: 165 0227 1                                           1 if only names are to be printed. This can
: 166 0228 1                                           occur if called from FOR$$UDF_RN to satisfy
: 167 0229 1                                           a "?" inquiry.
: 168 0230 1
: 169 0231 1 ! IMPLICIT INPUTS:
: 170 0232 1
: 171 0233 1     CCB                                       ! Register pointer to RAB/LUB/ISB
: 172 0234 1     ISB$_FMT_BEG                             ! Address of NAMELIST group descriptor
: 173 0235 1
: 174 0236 1 ! IMPLICIT OUTPUTS:
: 175 0237 1
: 176 0238 1
: 177 0239 1 ! COMPLETION STATUS:
: 178 0240 1
: 179 0241 1     NONE
: 180 0242 1
: 181 0243 1 ! SIDE EFFECTS:
: 182 0244 1
: 183 0245 1
: 184 0246 1 ! --
: 185 0247 1
: 186 0248 1 ! <BLF/PAGE>

```

188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244

```

0249 1  +-+
0250 1  Each NAMELIST descriptor block has the following form:
0251 1
0252 1      3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1
0253 1      1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
0254 1
0255 1  +-----+
0256 1  0  | Address of ASCII name of NAMELIST group |
0257 1  +-----+
0258 1  1  | Reserved          | Count of NAMELIST variables |
0259 1  +-----+
0260 1  2  | Address of ASCII name of variable 1 |
0261 1  +-----+
0262 1  3  | Address of standard VAX descriptor for variable 1 |
0263 1  +-----+
0264 1  4  |                                     |
0265 1  |                                     |
0266 1  5  | Address of ASCII name of variable n |
0267 1  +-----+
0268 1  6  | Address of standard VAX descriptor for variable n |
0269 1  +-----+
0270 1
0271 1
0272 1  The NAMELIST group name and the variable names which are pointed to in
0273 1  the NAMELIST descriptor block are upper case only. The FORTRAN
0274 1  compiler or other calling program is responsible for case conversion
0275 1  of the name strings. In NAMELIST input data, case is significant only
0276 1  in character literals. The run-time library is responsible for case
0277 1  conversion of NAMELIST input data.
0278 1
0279 1  The allowable data types in variable descriptors are BU (BYTE), WU,
0280 1  LU, W, L, F, D, G, H, T, FC, DC, and GC. The allowable descriptor
0281 1  classes are scalar and array. For the array class descriptor, the
0282 1  descriptor flags COLUMN, COEFF, and BOUNDS must be set, indicating
0283 1  column-major order and the presence of coefficient and bounds blocks.
0284 1  The number of dimensions must not exceed 7.
0285 1  --
0286 1
0287 1  !<BLF/PAGE>
0288 2  BEGIN
0289 2
0290 2  EXTERNAL REGISTER
0291 2  CCB = 11: REF $FOR$CCB_DECL;
0292 2
0293 2  LOCAL
0294 2  GROUP: REF VECTOR [, LONG],          ! NAMELIST group descriptor
0295 2  NVAR$,                                ! Number of variables in group
0296 2  VALUE_ADDR: REF VECTOR [, BYTE];     ! Address of value
0297 2
0298 2  !+
0299 2  ! Write out group name
0300 2  !-
0301 2
0302 2  GROUP = .CCB [ISB$A_FMT_BEG];
0303 2  VALUE_ADDR = .GROUP[0];              ! Address of group name counted string
0304 2  IF NOT CHECK_FIELD (.VALUE_ADDR [0] + 2) ! Include leading "$"
0305 2  THEN

```



```
245 0306 3 BEGIN
246 0307 3 FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
247 0308 3 RETURN;
248 0309 2 END;
249 0310 2 CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]); ! Write leading space
250 0311 2 CH$WCHAR_A (%C'$', CCB [LUB$A_BUF_PTR]); ! Write leading $
251 0312 2 CCB [LUB$A_BUF_PTR] = CH$MOVE(.VALUE_ADDR [0], VALUE_ADDR [1], .CCB [LUB$A_BUF_PTR]);
252 0313 2 FOR$$REC_WSN1 ?);
253 0314 2
254 0315 2 !+
255 0316 2 ! Scan through NAMELIST group and write all variables to the output stream.
256 0317 2 !-
257 0318 2
258 0319 2
259 0320 2 DECR NVAR$ FROM (. (GROUP [1]) <0,16,0> - 1) TO 0 DO
260 0321 2 BEGIN
261 0322 3 LOCAL
262 0323 3 OUT_NAME_LEN; ! Output name length
263 0324 3
264 0325 3
265 0326 3 GROUP = GROUP [2]; ! Skip to next variable
266 0327 3 VALUE_ADDR = .GROUP [0]; ! Address of variable name counted string
267 0328 3
268 0329 3 !+
269 0330 3 ! Compute output name length so that the names are padded to lengths
270 0331 3 ! of 7, 15, 23, etc. Add a space for before the '='.
271 0332 3 !-
272 0333 3
273 0334 3 OUT_NAME_LEN = .VALUE_ADDR [0];
274 0335 3 IF NOT .NAMES_ONLY
275 0336 3 THEN
276 0337 3 OUT_NAME_LEN = .OUT_NAME_LEN + (8 - (.VALUE_ADDR [0] MOD 8));
277 0338 3
278 0339 3 IF NOT CHECK_FIELD (.OUT_NAME_LEN + 2) ! Include leading space, trailing '='
279 0340 3 THEN
280 0341 4 BEGIN
281 0342 4 FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
282 0343 4 RETURN;
283 0344 3 END;
284 0345 3
285 0346 3 !+
286 0347 3 ! Write out variable name
287 0348 3 !-
288 0349 3
289 0350 3 CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
290 0351 3 CCB [LUB$A_BUF_PTR] = CH$COPY(.VALUE_ADDR [0], VALUE_ADDR [1],
291 0352 3 %C' ', .OUT_NAME_LEN, .CCB [LUB$A_BUF_PTR]);
292 0353 3
293 0354 3 !+
294 0355 3 ! Only print values if NAMES_ONLY is false.
295 0356 3 !-
296 0357 3
297 0358 3 IF NOT .NAMES_ONLY
298 0359 3 THEN
299 0360 4 BEGIN
300 0361 4 CH$WCHAR_A (%C'=', CCB [LUB$A_BUF_PTR]);
301 0362 4
```

```
302 0363 4      !+
303 0364 4      !- Output all values in variable
304 0365 4
305 0366 4
306 0367 5      BEGIN
307 0368 5      LOCAL
308 0369 5      VAR_DESC: REF BLOCK [, BYTE],      : Variable descriptor
309 0370 5      CUR_ADR,      : Current variable address
310 0371 5      END_ADR,      : End of variable
311 0372 5      ELEM_TYPE,      : Element datatype passed to FOR$$UDF_WL1
312 0373 5      CMPLX_FLAG;      : Complex flag passed to FOR$$UDF_WL1
313 0374 5      VAR_DESC = .GROUP [1];      ! Get descriptor
314 0375 5      CUR_ADR = .VAR_DESC [DSC$A_POINTER];
315 0376 5      IF .VAR_DESC [DSC$B_CLASS] EQL DSC$K_CLASS_A
316 0377 5      THEN
317 0378 5          END_ADR = .CUR_ADR + .VAR_DESC [DSC$L_ARSIZE]
318 0379 5      ELSE
319 0380 5          END_ADR = .CUR_ADR + .VAR_DESC [DSC$W_LENGTH];
320 0381 5      SELECTIONE .VAR_DESC [DSC$B_DTYPE] OF
321 0382 5      SET
322 0383 5          [DSC$K_DTYPE_FC]:
323 0384 6              BEGIN
324 0385 6                  ELEM_TYPE = DSC$K_DTYPE_F;
325 0386 6                  CMPLX_FLAG = 0;
326 0387 5                  END;
327 0388 5          [DSC$K_DTYPE_DC]:
328 0389 6              BEGIN
329 0390 6                  ELEM_TYPE = DSC$K_DTYPE_D;
330 0391 6                  CMPLX_FLAG = 0;
331 0392 5                  END;
332 0393 5          [DSC$K_DTYPE_GC]:
333 0394 6              BEGIN
334 0395 6                  ELEM_TYPE = DSC$K_DTYPE_G;
335 0396 6                  CMPLX_FLAG = 0;
336 0397 5                  END;
337 0398 5          [OTHERWISE]:
338 0399 6              BEGIN
339 0400 6                  ELEM_TYPE = .VAR_DESC [DSC$B_DTYPE];
340 0401 6                  !+
341 0402 6                  ! FORTRAN passes us BU for B, so change it here.
342 0403 6                  IF .ELEM_TYPE EQL DSC$K_DTYPE_BU
343 0404 6                  THEN
344 0405 6                      ELEM_TYPE = DSC$K_DTYPE_B;
345 0406 6                  CMPLX_FLAG = 2;
346 0407 5                  END;
347 0408 5          TES;
348 0409 5
349 0410 5      WHILE .END_ADR GTRA .CUR_ADR DO
350 0411 6          BEGIN
351 0412 6              LOCAL
352 0413 6                  CUR_POS,
353 0414 6                  REPEAT_COUNT;
354 0415 6              !+
355 0416 6              ! Build repeat count
356 0417 6              !-
357 0418 6
358 0419 6          REPEAT_COUNT = 1;
```

```
359 0420 6 CUR_POS = .CUR_ADR + .VAR_DESC [DSC$W_LENGTH];
360 0421 6 WHILE .CUR_POS - LSSA .END_ADR DO
361 0422 7 BEGIN
362 0423 7 IF NOT CH$EQL (.VAR_DESC [DSC$W_LENGTH],
363 0424 7 .CUR_ADR
364 0425 7 .VAR_DESC [DSC$W_LENGTH],
365 0426 7 .CUR_POS,
366 0427 7 0)
367 0428 7 THEN
368 0429 7 EXITLOOP;
369 0430 7 CUR_POS = .CUR_POS + .VAR_DESC [DSC$W_LENGTH];
370 0431 7 REPEAT_COUNT = .REPEAT_COUNT + 1;
371 0432 6 END;
372 0433 6
373 0434 6
374 0435 6 !+ Is this variable of type CHARACTER? If so, do all the
375 0436 6 processing here. Otherwise, let FOR$$UDF_WL1 do most of
376 0437 6 the work.
377 0438 6 !-
378 0439 6
379 0440 6 IF .ELEM_TYPE EQL DSC$K_DTYPE_T
380 0441 6 THEN
381 0442 7 BEGIN
382 0443 7 !+
383 0444 7 ! It's CHARACTER.
384 0445 7 !-
385 0446 7
386 0447 7 LOCAL
387 0448 7 REPEAT_DSC: DSC$DESCRIPTOR, ! Repeat string descriptor
388 0449 7 REPEAT_STR: VECTOR [12, BYTE], ! Repeat string
389 0450 7 FAO_DSC: DSC$DESCRIPTOR; ! FAO control string descr
390 0451 7 !+
391 0452 7 ! Build repeat count string.
392 0453 7 !-
393 0454 7
394 0455 7 REPEAT_DSC [DSC$W_LENGTH] = 0;
395 0456 7 IF .REPEAT_COUNT GTR 1
396 0457 7 THEN
397 0458 8 BEGIN
398 0459 8 REPEAT_DSC [DSC$W_LENGTH] = 12;
399 0460 8 REPEAT_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
400 0461 8 REPEAT_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
401 0462 8 REPEAT_DSC [DSC$A_POINTER] = REPEAT_STR;
402 0463 8 FAO_DSC [DSC$A_POINTER] = UPLIT BYTE ('!SL*');
403 0464 8 FAO_DSC [DSC$W_LENGTH] = %CHARCOUNT ('!SL*');
404 0465 8 FAO_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
405 0466 8 FAO_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
406 P 0467 8 $FAO (FAO_DSC, ! Control string
407 P 0468 8 REPEAT_DSC [DSC$W_LENGTH], ! Returned length
408 P 0469 8 REPEAT_DSC, ! Output string
409 0470 8 .REPEAT_COUNT);
410 0471 7 END;
411 0472 7
412 0473 7
413 0474 7 !+ See if there is enough room for the repeat count
414 0475 7 !-
415 0476 7
```

```
416 0477 7 IF NOT CHECK_FIELD (2 + . REPEAT_DSC [DSC$W_LENGTH])
417 0478 7 THEN
418 0479 8 BEGIN
419 0480 8 FOR$$REC_WSN1 ();
420 0481 8 IF NOT CHECK_FIELD (2 + .REPEAT_DSC [DSC$W_LENGTH])
421 0482 8 THEN
422 0483 8 FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
423 0484 7 END;
424 0485 7
425 0486 7 !+
426 0487 7 ! Write out a leading space, the repeat count and an
427 0488 7 ! initial apostrophe.
428 0489 7 !-
430 0491 7 CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
431 0492 7 CCB [LUB$A_BUF_PTR] = CH$MOVE (.REPEAT_DSC [DSC$W_LENGTH], REPEAT_STR,
432 0493 7 .CCB [LUB$A_BUF_PTR]);
433 0494 7 CH$WCHAR_A (%C''', CCB [LUB$A_BUF_PTR]);
434 0495 7
435 0496 7 !+
436 0497 7 ! Write out each character of the string, substituting two
437 0498 7 ! apostrophes for each apostrophe found in the string.
438 0499 7 !-
439 0500 7
440 0501 7 INCR I FROM 1 TO .VAR_DESC [DSC$W_LENGTH] DO
441 0502 8 BEGIN
442 0503 8 IF NOT CHECK_FIELD (1)
443 0504 8 THEN
444 0505 9 BEGIN
445 0506 9 FOR$$REC_WSN1 ();
446 0507 9 CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
447 0508 8 END;
448 0509 8 IF CH$RCHAR (.CUR_ADR) EQL %C''''
449 0510 8 THEN
450 0511 9 BEGIN
451 0512 9 CH$WCHAR_A (%C''''', CCB [LUB$A_BUF_PTR]);
452 0513 9 IF NOT CHECK_FIELD (1)
453 0514 9 THEN
454 0515 10 BEGIN
455 0516 10 FOR$$REC_WSN1 ();
456 0517 10 CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
457 0518 9 END;
458 0519 8 END;
459 0520 8 COPY_BYTE_A (CUR_ADR, CCB [LUB$A_BUF_PTR]);
460 0521 7 END;
461 0522 7
462 0523 7 !+
463 0524 7 ! Write out the closing apostrophe.
464 0525 7 !-
465 0526 7
466 0527 7 IF NOT CHECK_FIELD (1)
467 0528 7 THEN
468 0529 8 BEGIN
469 0530 8 FOR$$REC_WSN1 ();
470 0531 8 CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
471 0532 7 END;
472 0533 7 CH$WCHAR_A (%C''''', CCB [LUB$A_BUF_PTR]);
```

```

473 0534 7          CUR_ADR = .CUR_POS;
474 0535 7          END
475 0536 7
476 0537 6          ELSE
477 0538 6
478 0539 7          BEGIN
479 0540 7          | +
480 0541 7          | Not CHARACTER.
481 0542 7          | Call list directed routine to output value.
482 0543 7          | -
483 0544 7
484 0545 7          FOR$$UDF_WL1 (.ELEM_TYPE, .VAR_DESC [DSC$W_LENGTH], .CUR_ADR,
485 0546 7          | .CMPLX_FLAG, .REPEAT_COUNT);
486 0547 7          CUR_ADR = .CUR_POS;
487 0548 6          END;
488 0549 6
489 0550 6          | +
490 0551 6          | Put out a separating comma if values to come
491 0552 6          | -
492 0553 6
493 0554 6          IF .CUR_ADR LSSA .END_ADR
494 0555 6          THEN
495 0556 7          BEGIN
496 0557 7          IF NOT CHECK_FIELD (1)
497 0558 7          THEN
498 0559 8          BEGIN
499 0560 8          FOR$$REC_WSN1 ();
500 0561 8          CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
501 0562 7          END;
502 0563 7          CH$WCHAR_A (%C',', CCB [LUB$A_BUF_PTR]);
503 0564 6          END;
504 0565 5          END;
505 0566 4          END;
506 0567 4
507 0568 4          | +
508 0569 4          | If this is not the last variable, write out a comma
509 0570 4          | -
510 0571 4
511 0572 4          IF .NVAR$ NEQ 0
512 0573 4          THEN
513 0574 5          BEGIN
514 0575 5          IF NOT CHECK_FIELD (1)
515 0576 5          THEN
516 0577 6          BEGIN
517 0578 6          FOR$$REC_WSN1 ();
518 0579 6          CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
519 0580 5          END;
520 0581 5          CH$WCHAR_A (%C',', CCB [LUB$A_BUF_PTR]);
521 0582 4          END;
522 0583 4
523 0584 3          END;
524 0585 3
525 0586 3          | +
526 0587 3          | Write this record.
527 0588 3          | -
528 0589 3          FOR$$REC_WSN1 ();
529 0590 3

```

```

530 0591 3
531 0592 2
532 0593 2
533 0594 2
534 0595 2
535 0596 2
536 0597 2
537 0598 2
538 0599 2
539 0600 2
540 0601 2
541 0602 2
542 0603 2
543 0604 2
544 0605 2
545 0606 2
546 0607 2
547 0608 2
548 0609 1

```

```

      END;
      !+
      ! All variables written. Put out $END block delimiter.
      !-
      IF NOT CHECK_FIELD (5)
      THEN
      BEGIN
      FOR$$SIGNAL_STO (FOR$$K_OUTSTADVE);
      RETURN;
      END;
      [CCB [LUB$$A_BUF_PTR] = CH$MOVE (5, UPLIT BYTE (' $END'), .CCB [LUB$$A_BUF_PTR]);
      FOR$$REC_WSN1 (?);
      RETURN;
      END;

```

! End of routine

|  |       |    |      |      |    |       |        |                    |   |      |  |
|--|-------|----|------|------|----|-------|--------|--------------------|---|------|--|
|  | 44    | 2A | 4C   | 53   | 21 | 0000E | P.AAA: | .ASCII             | \\!SL*\   |      |  |
|  |       | 4E | 45   | 24   | 20 | 00012 | P.AAB: | .ASCII             | \\ \$END\                                       |      |  |
|  |       |    |      |      |    |       |        | .EXTRN             | SYSS\$FAO                                       |      |  |
|  |       |    |      |      |    | 07FC  | 00000  | .ENTRY             | FOR\$\$DO_NML_OUTPUT, Save R2,R3,R4,R5,R6,R7,-; | 0211 |  |
|  |       | 5E |      | 34   | C2 | 00002 |        | SUBL2              | R8,R9,R10                                       |      |  |
|  |       |    | FF7C | CB   | DD | 00005 |        | PUSHL              | #52, SP   |      |  |
|  |       | 5A | 00   | BE   | D0 | 00009 |        | MOVL               | -132(CCB)                                       | 0302 |  |
|  |       | 7E |      | 6A   | 9A | 0000D |        | MOVZBL             | @GROUP, VALUE_ADDR                              | 0303 |  |
|  |       | 6E |      | 02   | C0 | 00010 |        | ADDL2              | (VALUE_ADDR), -(SP)                             | 0304 |  |
|  | 0000V | CF |      | 01   | FB | 00013 |        | CALLS              | #2, (SP)  |      |  |
|  |       | 5D |      | 50   | E9 | 00018 |        | BLBC               | #1, CHECK_FIELD                                 |      |  |
|  |       | 56 | B0   | AB   | 9E | 0001B |        | MOVAB              | R0, 3\$   |      |  |
|  | 00    | B6 |      | 20   | 90 | 0001F |        | MOVAB              | -80(CCB), R6                                    | 0310 |  |
|  |       |    |      | 66   | D6 | 00023 |        | MOVB               | #32, @0(R6)                                     |      |  |
|  | 00    | B6 |      | 24   | 90 | 00025 |        | INCL               | (R6)  | 0311 |  |
|  |       |    |      | 66   | D6 | 00029 |        | MOVB               | #36, @0(R6)                                     |      |  |
|  |       |    |      | 6A   | 9A | 0002B |        | INCL               | (R6)  |      |  |
|  | 00    | B6 | 01   | 50   | 6A | 0002B |        | MOVZBL             | (VALUE_ADDR), R0                                | 0312 |  |
|  |       |    |      | 50   | 28 | 0002E |        | MOVZBL             | R0, 1(VALUE_ADDR), @0(R6)                       |      |  |
|  |       |    |      | 66   | D0 | 00034 |        | MOVC3              | R3, (R6)  |      |  |
|  |       |    |      | 53   | D0 | 00034 |        | MOVL               | R3, (R6)  |      |  |
|  |       |    |      | 00   | 16 | 00037 |        | JSB                | FOR\$\$REC_WSN1                                 | 0313 |  |
|  |       | 18 | AE   | 04   | AC | 0003D |        | MCOML              | FOR\$\$REC_WSN1                                 | 0335 |  |
|  |       |    |      | 04   | AC | 0003D |        | NAMES ONLY, 24(SP) |   |      |  |
|  |       | 7E |      | 04   | C1 | 00042 |        | ADDL3              | #4, GROUP, -(SP)                                |      |  |
|  |       |    |      | 9E   | 3C | 00046 |        | MOVZWL             | @(SP)+, NVAR\$                                  |      |  |
|  |       |    |      | 0231 | 31 | 0004A |        | BRW                | 32\$  |      |  |
|  |       |    |      | 08   | C0 | 0004D | 1\$:   | ADDL2              | #8, GROUP                                       | 0326 |  |
|  |       | 6E |      | 08   | C0 | 0004D |        | MOVL               | @GROUP, VALUE_ADDR                              | 0327 |  |
|  |       | 5A | 00   | BE   | D0 | 00050 |        | MOVZBL             | (VALUE_ADDR), -OUT_NAME_LEN                     | 0334 |  |
|  |       | 52 |      | 6A   | 9A | 00054 |        | BLBC               | 24(SP), 2\$                                     | 0335 |  |
|  |       | 15 |      | AE   | E9 | 00057 |        | MOVZBL             | (VALUE_ADDR), R0                                | 0337 |  |
|  |       |    |      | 6A   | 9A | 0005B |        | EMUL               | #1, R0, #0, -(SP)                               |      |  |
|  | 7E    |    | 00   | 50   | 01 | 7A    | 0005E  | EDIV               | #8, (SP)+, R0, R0                               |      |  |
|  |       |    |      | 50   | 08 | 7B    | 00063  | SUBL3              | R0, OUT_NAME_LEN, R0                            |      |  |
|  | 50    |    | 50   | 52   | 50 | C3    | 00068  |                    |   |      |  |

|    |       |    |    |    |    |       |        |   |                            |                                 |
|----|-------|----|----|----|----|-------|--------|---|----------------------------|---------------------------------|
|    |       | 52 | 08 | A0 | 9E | 0006C | MOVAB  | 8(R0), OUT_NAME_LEN                     |                            |                                 |
|    |       |    | 02 | A2 | 9F | 00070 | PUSHAB | 2(OUT_NAME_LEN)                         | 0339                       |                                 |
|    | 0000V | CF |    | 01 | FB | 00073 | CALLS  | #1, CHECK_FIELD                         |                            |                                 |
|    |       | 03 |    | 50 | E8 | 00078 | BLBS   | R0, 4\$                                 |                            |                                 |
|    |       |    |    | 02 | 13 | 31    | 0007B  | BRW                                     | 35\$                       |                                 |
|    |       | 56 | B0 | AB | 9E | 0007E | MOVAB  | -80(CCB), R6                            | 0350                       |                                 |
|    | 00    | B6 |    | 20 | 90 | 00082 | MOVAB  | #32, @0(R6)                             |                            |                                 |
|    |       |    |    | 66 | D6 | 00086 | INCL   | (R6)                                    |                            |                                 |
| 52 |       | 50 |    | 6A | 9A | 00088 | MOVZBL | (VALUE_ADDR), R0                        | 0351                       |                                 |
|    | 20    | 01 | AA | 50 | 2C | 0008B | MOVCS  | R0, 1(VALUE_ADDR), #32, OUT_NAME_LEN, - | 0352                       |                                 |
|    |       |    |    | 00 | 66 | 00091 |        | @0(R6)                                  |                            |                                 |
|    |       | 66 |    | 53 | D0 | 00093 | MOVL   | R3, (R6)                                |                            |                                 |
|    |       | 03 |    | 18 | AE | 00096 | BLBS   | 24(SP), 5\$                             | 0358                       |                                 |
|    |       |    |    | 01 | DB | 31    | 0009A  | BRW                                     | 31\$                       |                                 |
|    | 00    | B6 |    | 3D | 90 | 0009D | MOVAB  | #61, @0(R6)                             | 0361                       |                                 |
|    |       |    |    | 66 | D6 | 000A1 | INCL   | (R6)                                    |                            |                                 |
|    | 50    | 6E |    | 04 | C1 | 000A3 | ADDL3  | #4, GROUP, R0                           | 0374                       |                                 |
|    |       | 57 |    | 60 | D0 | 000A7 | MOVL   | (R0), VAR_DESC                          |                            |                                 |
|    |       | 59 |    | 04 | A7 | 000AA | MOVL   | 4(VAR_DESC), CUR_ADR                    | 0375                       |                                 |
|    |       | 04 |    | 03 | A7 | 000AE | CMPB   | 3(VAR_DESC), #4                         | 0376                       |                                 |
|    |       |    |    | 08 | 12 | 000B2 | BNEQ   | 6\$                                     |                            |                                 |
|    | 10    | AE |    | 0C | B7 | 49    | 9E     | 000B4                                   | MOVAB                      | @12(VAR_DESC)[CUR_ADR], END_ADR |
|    |       |    |    | 08 | 11 | 000BA | BRB    | 7\$                                     | 0378                       |                                 |
|    |       | 50 |    | 67 | 3C | 000BC | MOVZWL | (VAR_DESC), R0                          | 0380                       |                                 |
| 10 | AE    | 59 |    | 50 | C1 | 000BF | ADDL3  | R0, CUR_ADR, END_ADR                    |                            |                                 |
|    |       | 50 |    | 02 | A7 | 9A    | 000C4  | MOVZBL                                  | 2(VAR_DESC), R0            | 0381                            |
|    |       | 0C |    | 50 | 91 | 000C8 | CMPB   | R0, #T2                                 | 0383                       |                                 |
|    |       |    |    | 05 | 12 | 000CB | BNEQ   | 8\$                                     |                            |                                 |
|    |       | 58 |    | 0A | D0 | 000CD | MOVL   | #10, ELEM_TYPE                          | 0385                       |                                 |
|    |       |    |    | 12 | 11 | 000D0 | BRB    | 10\$                                    | 0386                       |                                 |
|    |       | 0D |    | 50 | 91 | 000D2 | CMPB   | R0, #13                                 | 0388                       |                                 |
|    |       |    |    | 05 | 12 | 000D5 | BNEQ   | 9\$                                     |                            |                                 |
|    |       | 58 |    | 0B | D0 | 000D7 | MOVL   | #11, ELEM_TYPE                          | 0390                       |                                 |
|    |       |    |    | 08 | 11 | 000DA | BRB    | 10\$                                    | 0391                       |                                 |
|    |       | 1D |    | 50 | 91 | 000DC | CMPB   | R0, #29                                 | 0393                       |                                 |
|    |       |    |    | 08 | 12 | 000DF | BNEQ   | 11\$                                    |                            |                                 |
|    |       | 58 |    | 1B | D0 | 000E1 | MOVL   | #27, ELEM_TYPE                          | 0395                       |                                 |
|    |       |    |    | 0C | AE | D4    | 000E4  | CLRL                                    | CMPLX_FLAG                 | 0396                            |
|    |       |    |    | 0F | 11 | 000E7 | BRB    | 13\$                                    | 0381                       |                                 |
|    |       | 58 |    | 50 | D0 | 000E9 | MOVL   | R0, ELEM_TYPE                           | 0400                       |                                 |
|    |       | 02 |    | 58 | D1 | 000EC | CMPL   | ELEM_TYPE, #2                           | 0403                       |                                 |
|    |       |    |    | 03 | 12 | 000EF | BNEQ   | 12\$                                    |                            |                                 |
|    |       | 58 |    | 06 | D0 | 000F1 | MOVL   | #6, ELEM_TYPE                           | 0405                       |                                 |
|    | 0C    | AE |    | 02 | D0 | 000F4 | MOVL   | #2, CMPLX_FLAG                          | 0406                       |                                 |
|    |       | 59 |    | 10 | AE | D1    | 000F8  | CMPL                                    | END_ADR, CUR_ADR           | 0410                            |
|    |       |    |    | 03 | 1A | 000FC | BGTRU  | 14\$                                    |                            |                                 |
|    |       |    |    | 01 | 54 | 31    | 000FE  | BRW                                     | 29\$                       |                                 |
|    |       | 54 |    | 01 | D0 | 00101 | MOVL   | #1, REPEAT_COUNT                        | 0419                       |                                 |
|    | 04    | AE |    | 67 | 3C | 00104 | MOVZWL | (VAR_DESC), 4(SP)                       | 0420                       |                                 |
|    | 08    | AE |    | 04 | BE | 49    | 9E     | 00108                                   | MOVAB                      | @4(SP)[CUR_ADR], CUR_POS        |
|    | 10    | AE |    | 08 | AE | D1    | 0010E  | CMPL                                    | CUR_POS, END_ADR           | 0421                            |
|    |       |    |    | 11 | 1E | 00113 | BGEQU  | 16\$                                    |                            |                                 |
| 08 | BE    |    |    | 04 | AE | 29    | 00115  | CMPC3                                   | 4(SP), (CUR_ADR), @CUR_POS | 0423                            |
|    |       | 69 |    | 09 | 12 | 0011B | BNEQ   | 16\$                                    |                            |                                 |
|    |       |    |    | 08 | AE | C0    | 0011D  | ADDL2                                   | 4(SP), CUR_POS             | 0430                            |
|    |       |    |    | 54 | D6 | 00122 | INCL   | REPEAT_COUNT                            | 0431                       |                                 |
|    |       |    |    | E8 | 11 | 00124 | BRB    | 15\$                                    | 0421                       |                                 |

|    |           |    |           |    |       |       |        |                                |      |
|----|-----------|----|-----------|----|-------|-------|--------|--------------------------------|------|
|    | OE        |    | 58        | D1 | 00126 | 16\$: | CMPL   | ELEM_TYPE, #14                 | 0440 |
|    |           |    | 03        | 13 | 00129 |       | BEQL   | 17\$                           |      |
|    |           |    | 00E9      | 31 | 0012B |       | BRW    | 25\$                           |      |
|    |           | 30 | AE        | B4 | 0012E | 17\$: | CLRW   | REPEAT_DSC                     | 0455 |
|    | 01        |    | 54        | D1 | 00131 |       | CMPL   | REPEAT_COUNT, #1               | 0456 |
|    |           |    | 2D        | 15 | 00134 |       | BLEQ   | 18\$                           |      |
|    | 30        | AE | 010E000C  | 8F | D0    | 00136 | MOVL   | #17694732, REPEAT_DSC          | 0457 |
|    | 34        | AE | 24        | AE | 9E    | 0013E | MOVAB  | REPEAT_STR, REPEAT_DSC+4       | 0462 |
|    | 20        | AE | FEB0      | CF | 9E    | 00143 | MOVAB  | P.AAA, -FAO_DSC+4              | 0463 |
|    | 1C        | AE | 010E0004  | 8F | D0    | 00149 | MOVL   | #17694724, -FAO_DSC            | 0464 |
|    |           |    | 54        | DD | 00151 |       | PUSHL  | REPEAT_COUNT                   | 0470 |
|    |           | 34 | AE        | 9F | 00153 |       | PUSHAB | REPEAT_DSC                     |      |
|    |           | 38 | AE        | 9F | 00156 |       | PUSHAB | REPEAT_DSC                     |      |
|    |           | 28 | AE        | 9F | 00159 |       | PUSHAB | FAO_DSC                        |      |
|    | 00000000G | 00 | 04        | FB | 0015C |       | CALLS  | #4, -SYSSFAO                   |      |
|    |           | 52 | 30        | AE | 3C    | 00163 | MOVZWL | REPEAT_DSC, R2                 | 0477 |
|    |           | 52 |           | 02 | CO    | 00167 | ADDL2  | #2, R2                         |      |
|    |           |    | 52        | DD | 0016A |       | PUSHL  | R2                             |      |
|    | 0000V     | CF |           | 01 | FB    | 0016C | CALLS  | #1, CHECK_FIELD                |      |
|    |           | 1B |           | 50 | E8    | 00171 | BLBS   | R0, 19\$                       |      |
|    |           |    | 00000000G | 00 | 16    | 00174 | JSB    | FORSSREC_WSN1                  | 0480 |
|    |           |    |           | 52 | DD    | 0017A | PUSHL  | R2                             | 0481 |
|    | 000GV     | CF |           | 01 | FB    | 0017C | CALLS  | #1, CHECK_FIELD                |      |
|    |           | 0B |           | 50 | E8    | 00181 | BLBS   | R0, 19\$                       |      |
|    |           | 7E | 42        | 8F | 9A    | 00184 | MOVZBL | #66, -(SP)                     | 0483 |
|    | 00000000G | 00 |           | 01 | FB    | 00188 | CALLS  | #1, FORSSIGNAL_STO             |      |
|    |           | 56 | 80        | AB | 9E    | 0018F | MOVAB  | -80(CCB), R6                   | 0491 |
|    | 00        | B6 |           | 20 | 90    | 00193 | MOVB   | #32, @0(R6)                    |      |
|    |           |    |           | 66 | D6    | 00197 | INCL   | (R6)                           |      |
| 00 | B6        | 24 | AE        | 30 | AE    | 28    | MOVCS  | REPEAT_DSC, REPEAT_STR, @0(R6) | 0493 |
|    |           | 66 |           | 53 | D0    | 001A0 | MOVL   | R3, (R6)                       |      |
|    |           | 00 | B6        | 27 | 90    | 001A3 | MOVB   | #39, @0(R6)                    | 0494 |
|    |           |    |           | 66 | D6    | 001A7 | INCL   | (R6)                           |      |
|    |           |    |           | 52 | D6    | 001A9 | CLRL   | I                              | 0501 |
|    |           |    |           | 45 | 11    | 001AB | BRB    | 23\$                           |      |
|    |           |    |           | 01 | 1D    | 001AD | PUSHL  | #1                             | 0503 |
|    | 0000V     | CF |           | 01 | FB    | 001AF | CALLS  | #1, CHECK_FIELD                |      |
|    |           | 0D |           | 50 | E8    | 001B4 | BLBS   | R0, 21\$                       |      |
|    |           |    | 00000000G | 00 | 16    | 001B7 | JSB    | FORSSREC_WSN1                  | 0506 |
|    | B0        | BB |           | 20 | 90    | 001BD | MOVB   | #32, @-80(CCB)                 | 0507 |
|    |           |    | 80        | AB | D6    | 001C1 | INCL   | -80(CCB)                       |      |
|    |           | 27 |           | 69 | 91    | 001C4 | CMPB   | (CUR_ADR), #39                 | 0509 |
|    |           |    |           | 1E | 12    | 001C7 | BNEQ   | 22\$                           |      |
|    | B0        | BB |           | 27 | 90    | 001C9 | MOVB   | #39, @-80(CCB)                 | 0512 |
|    |           |    | 80        | AB | D6    | 001CD | INCL   | -80(CCB)                       |      |
|    |           |    |           | 01 | 1D    | 001D0 | PUSHL  | #1                             | 0513 |
|    | 0000V     | CF |           | 01 | FB    | 001D2 | CALLS  | #1, CHECK_FIELD                |      |
|    |           | 0D |           | 50 | E8    | 001D7 | BLBS   | R0, 22\$                       |      |
|    |           |    | 00000000G | 00 | 16    | 001DA | JSB    | FORSSREC_WSN1                  | 0516 |
|    | B0        | BB |           | 20 | 90    | 001E0 | MOVB   | #32, @-80(CCB)                 | 0517 |
|    |           |    | 80        | AB | D6    | 001E4 | INCL   | -80(CCB)                       |      |
|    |           |    | 80        | AB | D6    | 001E7 | INCL   | -80(CCB)                       | 0520 |
|    |           | 50 | 80        | AB | D0    | 001EA | MOVL   | -80(CCB), R0                   |      |
|    | FF        | A0 |           | 89 | 90    | 001EE | MOVB   | (CUR_ADR)+-1(R0)               |      |
| B6 |           | 52 | 04        | AE | F3    | 001F2 | AOBLEQ | 4(SPT), I, 20\$                | 0501 |
|    |           |    |           | 01 | DD    | 001F7 | PUSHL  | #1                             | 0527 |
|    | 0000V     | CF |           | 01 | FB    | 001F9 | CALLS  | #1, CHECK_FIELD                |      |



|           |    |           |      |       |       |        |                       |                      |      |
|-----------|----|-----------|------|-------|-------|--------|-----------------------|----------------------|------|
|           | OD |           | 50   | E8    | 001FE | BLBS   | RO, 24\$              |                      |      |
|           |    | 00000000G | 00   | 16    | 00201 | JSB    | FOR\$\$REC_WSN1       | 0530                 |      |
| B0        | BB |           | 20   | 90    | 00207 | MOVB   | #32, @-80(CCB)        | 0531                 |      |
|           |    | B0        | AB   | D6    | 0020B | INCL   | -80(CCB)              |                      |      |
| B0        | BB |           | 27   | 90    | 0020E | MOVB   | #39, @-80(CCB)        | 0533                 |      |
|           |    | B0        | AB   | D6    | 00212 | INCL   | -80(CCB)              |                      |      |
|           |    |           | 13   | 11    | 00215 | BRB    | 26\$                  | 0534                 |      |
|           |    |           | 54   | DD    | 00217 | PUSHL  | REPEAT_COUNT          | 0546                 |      |
|           |    | 10        | AE   | DD    | 00219 | PUSHL  | CMPLX_FLAG            |                      |      |
|           |    |           | 59   | DD    | 0021C | PUSHL  | CUR_ADR               | 0545                 |      |
|           |    | 10        | AE   | DD    | 0021E | PUSHL  | 16(SP)                |                      |      |
|           |    |           | 58   | DD    | 00221 | PUSHL  | ELEM_TYPE             |                      |      |
| 00000000G | 00 |           | 05   | FB    | 00223 | CALLS  | #5, FOR\$\$UDF_WL1    |                      |      |
|           | 59 | 08        | AE   | D0    | 0022A | MOVL   | CUR_POS, CUR_ADR      | 0547                 |      |
| 10        | AE |           | 59   | D1    | 0022E | CMPL   | CUR_ADR, END_ADR      | 0554                 |      |
|           |    |           | 1E   | 1E    | 00232 | BGEQU  | 28\$                  |                      |      |
|           |    |           | 01   | DD    | 00234 | PUSHL  | #1                    | 0557                 |      |
| 0000V     | CF |           | 01   | FB    | 00236 | CALLS  | #1, CHECK_FIELD       |                      |      |
|           | OD |           | 50   | E8    | 0023B | BLBS   | RO, 27\$              |                      |      |
|           |    | 00000000G | 00   | 16    | 0023E | JSB    | FOR\$\$REC_WSN1       | 0560                 |      |
| B0        | BB |           | 20   | 90    | 00244 | MOVB   | #32, @-80(CCB)        | 0561                 |      |
|           |    | B0        | AB   | D6    | 00248 | INCL   | -80(CCB)              |                      |      |
| B0        | BB |           | 2C   | 90    | 0024B | MOVB   | #44, @-80(CCB)        | 0563                 |      |
|           |    | B0        | AB   | D6    | 0024F | INCL   | -80(CCB)              |                      |      |
|           |    |           | FEA3 | 31    | 00252 | BRW    | 13\$                  | 0410                 |      |
|           |    | 14        | AE   | D5    | 00255 | TSTL   | NVARS                 | 0572                 |      |
|           |    |           | 1E   | 13    | 00258 | BEQL   | 31\$                  |                      |      |
|           |    |           | 01   | DD    | 0025A | PUSHL  | #1                    | 0575                 |      |
| 0000V     | CF |           | 01   | FB    | 0025C | CALLS  | #1, CHECK_FIELD       |                      |      |
|           | OD |           | 50   | E8    | 00261 | BLBS   | RO, 30\$              |                      |      |
|           |    | 00000000G | 00   | 16    | 00264 | JSB    | FOR\$\$REC_WSN1       | 0578                 |      |
| B0        | BB |           | 20   | 90    | 0026A | MOVB   | #32, @-80(CCB)        | 0579                 |      |
|           |    | B0        | AB   | D6    | 0026E | INCL   | -80(CCB)              |                      |      |
| B0        | BB |           | 2C   | 90    | 00271 | MOVB   | #44, @-80(CCB)        | 0581                 |      |
|           |    | B0        | AB   | D6    | 00275 | INCL   | -80(CCB)              |                      |      |
|           |    | 00000000G | 00   | 16    | 00278 | JSB    | FOR\$\$REC_WSN1       | 0590                 |      |
|           | 02 | 14        | AE   | F4    | 0027E | SOBGEQ | NVARS, 33\$           | 0320                 |      |
|           |    |           | 03   | 11    | 00282 | BRB    | 34\$                  |                      |      |
|           |    |           | FDC6 | 31    | 00284 | BRW    | 1\$                   |                      |      |
|           |    |           | 05   | DD    | 00287 | PUSHL  | #5                    | 0598                 |      |
| 0000V     | CF |           | 01   | FB    | 00289 | CALLS  | #1, CHECK_FIELD       |                      |      |
|           | OC |           | 50   | E8    | 0028E | BLBS   | RO, 36\$              |                      |      |
|           | 7E | 42        | 8F   | 9A    | 00291 | MOVZBL | #66, -(SP)            | 0601                 |      |
| 00000000G | 00 |           | 01   | FB    | 00295 | CALLS  | #1, FOR\$\$SIGNAL_STO |                      |      |
|           |    |           | 04   | 0029C | RET   |        |                       | 0600                 |      |
| B0        | BB | FD59      | CF   | 05    | 28    | 0029D  | MOV3                  | #5, P.AAB, @-80(CCB) | 0604 |
|           |    | B0        | AB   | 53    | D0    | 002A4  | MOVL                  | R3, -80(CCB)         |      |
|           |    | 00000000G | 00   | 16    | 002A8 | JSB    | FOR\$\$REC_WSN1       | 0605                 |      |
|           |    |           | 04   | 002AE | RET   |        |                       | 0609                 |      |

; Routine Size: 687 bytes, Routine Base: \_FOR\$CODE + 0017

; 549 0610 1 !<BLF/PAGE>

```

: 551 0611 1 %SBTTL 'FOR$$UDF_WN9 - End WRITE NAMELIST'
: 552 0612 1 GLOBAL ROUTINE FOR$$UDF_WN9: JSB_UDF9 NOVALUE      ! End WRITE NAMELIST
: 553 0613 1 =
: 554 0614 1
: 555 0615 1
: 556 0616 1 ++
: 557 0617 1 |
: 558 0618 1 |   FUNCTIONAL DESCRIPTION:
: 559 0619 1 |   End a namelist-directed WRITE statement. This procedure, although
: 560 0620 1 |   a no-op, is necessary because FOR$IO_END dispatches to a UDF9 routine
: 561 0621 1 |   based on the statement type.
: 562 0622 1 |
: 563 0623 1 |   CALLING SEQUENCE:
: 564 0624 1 |   JSB FOR$$UDF_WN9
: 565 0625 1 |
: 566 0626 1 |   FORMAL PARAMETERS:
: 567 0627 1 |
: 568 0628 1 |   NONE
: 569 0629 1 |
: 570 0630 1 |   IMPLICIT INPUTS:
: 571 0631 1 |
: 572 0632 1 |   CCB                      ! Register pointer to RAB/LUB/ISB
: 573 0633 1 |
: 574 0634 1 |   IMPLICIT OUTPUTS:
: 575 0635 1 |
: 576 0636 1 |   NONE
: 577 0637 1 |
: 578 0638 1 |   COMPLETION STATUS: (or ROUTINE VALUE:)
: 579 0639 1 |
: 580 0640 1 |   NONE
: 581 0641 1 |
: 582 0642 1 |   SIDE EFFECTS:
: 583 0643 1 |
: 584 0644 1 |   NONE
: 585 0645 1 |
: 586 0646 1 | --
: 587 0647 1 |
: 588 0648 2 |   BEGIN
: 589 0649 2 |
: 590 0650 2 |   RETURN;
: 591 0651 2 |
: 592 0652 1 |   END;                      ! End of routine FOR$$UDF_WN9

```

05 0000 FOR\$\$UDF\_WN9::  
RSB

; 0652

: Routine Size: 1 bytes, Routine Base: \_FOR\$CODE + 02C6

: 593 0653 1 !<BLF/PAGE>

```

595 0654 1 %SBTTL 'CHECK_FIELD - Check field remaining for width'
596 0655 1 ROUTINE CHECK_FIELD (
597 0656 1     WIDTH
598 0657 1     ): CALL_CCB
599 0658 1     =
600 0659 1
601 0660 1  +-+
602 0661 1  FUNCTIONAL DESCRIPTION:
603 0662 1
604 0663 1      Determine if there are sufficient characters remaining in the current
605 0664 1      record for a field of a specified width.
606 0665 1
607 0666 1  CALLING SEQUENCE:
608 0667 1
609 0668 1      status = CHECK_FIELD (width.rl.v)
610 0669 1
611 0670 1  FORMAL PARAMETERS:
612 0671 1
613 0672 1      width  - The width of the field you wish to use
614 0673 1
615 0674 1  IMPLICIT INPUTS:
616 0675 1
617 0676 1      CCB                ! Register pointer to RAB/LUB/ISB
618 0677 1
619 0678 1  IMPLICIT OUTPUTS:
620 0679 1
621 0680 1      NONE
622 0681 1
623 0682 1  COMPLETION STATUS: (or ROUTINE VALUE:)
624 0683 1
625 0684 1      1 if the field will fit, 0 otherwise
626 0685 1
627 0686 1  SIDE EFFECTS:
628 0687 1
629 0688 1      NONE
630 0689 1
631 0690 1  --
632 0691 1
633 0692 2  BEGIN
634 0693 2
635 0694 2  EXTERNAL REGISTER
636 0695 2      CCB = 11: REF $FOR$CCB_DECL;
637 0696 2
638 0697 2  RETURN ((.CCB [LUB$A_BUF_PTR] + .WIDTH) LEQA .CCB [LUB$A_BUF_END]);
639 0698 2
640 0699 1  END;                                ! End of routine CHECK_FIELD

```

```

0000 0000 CHECK_FIELD:
51      B0  AB      04  AC  C1 00002      .WORD  Save nothing
          50  D4 00008      ADDL3  WIDTH, -80(CCB), R1
          51  D1 0000A      CLRL   R0
          02  1A 0000E      CMPL  R1, -76(CCB)
          BGTRU  1$

```

```

: 0655
: 0697
:
:

```

FORSSUDF\_WN  
1-005

FORSSUDF\_WN - FORTRAN WRITE NAMELIST UDF Level <sup>K 1</sup> 16-Sep-1984 00:53:55  
CHECK\_FIELD - Check field remaining for width 14-Sep-1984 12:32:56

VAX-11 Bliss-32 V4.0-742  
[FORRTL.SRC]FORUDFWN.B32;1

Page 18  
(7)

50 D6 00010 INCL R0  
04 0001? 1s: RET

: 0699

: Routine Size: 19 bytes, Routine Base: \_FOR\$CODE + 02C7

: 641 0700 1 !<BLF/PAGE>

FO  
1-

```

: 643      0701 1 END
: 644      0702 1
: 645      0703 0 ELUDOM
! End of module FOR$$UDF_WN
    
```

PSECT SUMMARY

```

Name          Bytes          Attributes
_FOR$CODE     730 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
    
```

Library Statistics

| File                                   | Total | Symbols Loaded | Percent | Pages Mapped | Processing Time |
|--|-------|----------------|---------|--------------|-----------------|
| _\$255\$DUA28:[SYSLIB]STARLET.L32;1    | 9776  | 17             | 0       | 581          | 00:01.1         |
| _\$255\$DUA28:[FORRTL.OBJ]FORLIB.L32;1 | 711   | 186            | 26      | 52           | 00:00.6         |
| _\$255\$DUA28:[FORRTL.OBJ]RTLLIB.L32;1 | 36    | 0              | 0       | 8            | 00:00.1         |

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:FORUDFWN/OBJ=OBJ\$:FORUDFWN MSRC\$:FORUDFWN/UPDATE=(ENHS:FORUDFWN)

```

: Size:          721 code + 9 data bytes
: Run Time:      00:18.6
: Elapsed Time: 00:53.2
: Lines/CPU Min: 2264
: Lexemes/CPU-Min: 13809
: Memory Used:  274 pages
: Compilation Complete
    
```







