



```

FFFFFFFFF 000000 RRRRRRRR UU UU DDDDDDDD FFFFFFFF WW WW UU UU
FFFFFFFFF 000000 RRRRRRRR UU UU DDDDDDDD FFFFFFFF WW WW UU UU
FF 00 00 RR RR UU UU DD DD FF WW WW UU UU
FF 00 00 RR RR UU UU DD DD FF WW WW UU UU
FF 00 00 RR RR UU UU DD DD FF WW WW UU UU
FFFFFFFFF 00 00 RRRRRRRR UU UU DD DD FFFFFFFF WW WW UU UU
FFFFFFFFF 00 00 RRRRRRRR UU UU DD DD FFFFFFFF WW WW UU UU
FF 00 00 RR RR UU UU DD DD FF WW WW UU UU
FF 00 00 RR RR UU UU DD DD FF WW WW UU UU
FF 00 00 RR RR UU UU DD DD FF WW WW UU UU
FF 000000 RR RR UUUUUUUUU DDDDDDDD FF FF UUUUUUUUUU
FF 000000 RR RR UUUUUUUUU DDDDDDDD FF FF UUUUUUUUUU

```

```

LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

```

1 0001 0 MODULE FOR$$UDF_WU ( ! FORTRAN Write unformatted UDF
2 0002 0 IDENT = '1-012' ! File: FORUDFWU.B32 Edit: JAW1012
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 FACILITY: FORTRAN Support Library - not user callable
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module implements FORTRAN Write unformatted I/O
35 0035 1 statements (sequential access - S, direct access - D,
36 0036 1 at the User data formatter level of
37 0037 1 abstraction (UDF level is 2nd level). This module
38 0038 1 calls the appropriate write record
39 0039 1 routine at the record handling level of abstraction (REC
40 0040 1 level is 3rd level) to write a record.
41 0041 1
42 0042 1 ENVIRONMENT: User access mode; reentrant AST level or not.
43 0043 1
44 0044 1 AUTHOR: Thomas N. Hastings; CREATION DATE: 20-Feb-77
45 0045 1
46 0046 1 MODIFIED BY:
47 0047 1 Thomas N. Hastings, 12-Mar-77: Version 01
48 0048 1 01 - original
49 0049 1 0-4 - Removed parameters to record level routines JMT 17-OCT-77
50 0050 1 0-05 - Use FOR$K_abcnmo5yz as EOSTERNAL LITERALS. TNH 27-Oct-77
51 0051 1 0-06 - Global register CCB. JMT 8-Apr-78
52 0052 1 0-07 - Change REQUIRE files for VAX system build. DGP 28-Apr-78
53 0053 1 0-08 - Use JSB linkages. TNH 22-May-78
54 0054 1 0-C; - PIC dispatch tables. TNH 7-June-78
55 0055 1 0-10 - Change file name to FORUDFWU.d32, and change the names of the
56 0056 1 REQUIRE files similarly. JBS 14-NOV-78
57 0057 1 1-001 - Update version number and copyright notice. JBS 16-NOV-78

```

```
58 0058 1 | 1-002 - Change REQUIRE file names from FOR... to OTS... JBS 06-DEC-78
59 0059 1 | 1-003 - Change references to I/O dispatch tables. DGP 08-Dec-78
60 0060 1 | 1-004 - Change dispatch tables references to longwords. DGP 11-Dec-78
61 0061 1 | 1-005 - Change ISB$A BUF_PTR, BUF_BEG, BUF_END, BUF_HIGH to LUB. DGP 08-Jan-79
62 0062 1 | 1-006 - Use 32-bit addresses for externals. JBS 27-JAN-1979
63 0063 1 | 1-007 - Use language-specific dispatch tables. JBS 26-JUN-1979
64 0064 1 | 1-008 - Use ISB dispatch offset symbols. SBL 12-July-1979
65 0065 1 | 1-009 - Signal an error if the record overflows and its not
66 0066 1 | segmented. V2 FT1 Report #61 SBL 18-Sept-1979
67 0067 1 | 1-010 - Implement unbuffered transfers for single-element lists: if
68 0068 1 | recordtype is not segmented and record size permits, set up
69 0069 1 | RAB so record will be written directly from element, and do
70 0070 1 | not copy element to buffer. JAW 06-May-1981
71 0071 1 | 1-011 - Continuation of 1-010. Simplify tests in FOR$$UDF_WU1.
72 0072 1 | JAW 02-Jun-1981
73 0073 1 | 1-012 - Continuation of 1-010. Recast test in FOR$$UDF_WU1 for better
74 0074 1 | code. Remove unnecessary resetting of flag in FOR$$UDF_WU9.
75 0075 1 | JAW 06-Jun-1981
76 0076 1 | --
77 0077 1 |
78 0078 1 | <BLF/PAGE>
```

```
80 0079 1 |
81 0080 1 | SWITCHES:
82 0081 1 |
83 0082 1 |
84 0083 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
85 0084 1 |
86 0085 1 |
87 0086 1 | LINKAGES:
88 0087 1 |
89 0088 1 |
90 0089 1 REQUIRE 'RTLIN:OTSLNK'; ! Define all linkages
91 0518 1 |
92 0519 1 |
93 0520 1 | TABLE OF CONTENTS:
94 0521 1 |
95 0522 1 |
96 0523 1 FORWARD ROUTINE
97 0524 1     FOR$$UDF_WU0 : JSB_UDF0 NOVALUE, ! initialization
98 0525 1     FOR$$UDF_WU1 : CALC_CCB NOVALUE, ! format one user I/O list element
99 0526 1     FOR$$UDF_WU9 : JSB_UDF9 NOVALUE; ! end of user I/O list - finish
100 0527 1 |
101 0528 1 |
102 0529 1 | INCLUDE FILES:
103 0530 1 |
104 0531 1 |
105 0532 1 REQUIRE 'RTLML:FORERR'; ! FORTRAN error codes
106 0600 1 REQUIRE 'RTLML:OTSISB'; ! I/O statement block (ISB) offsets
107 0768 1 |
108 0769 1 REQUIRE 'RTLML:OTSLUB'; ! Logical unit block (LUB)
109 0909 1 |
110 0910 1 REQUIRE 'RTLIN:OTSMAC'; ! Macros
111 1104 1 |
112 1105 1 REQUIRE 'RTLIN:RTLPSECT'; ! Define DELCARE_PSECTS macro
113 1200 1 |
114 1201 1 REQUIRE 'RTLIN:RTLDBG'; ! RTL debugging macros
115 1310 1 |
116 1311 1 LIBRARY 'RTLSTARLE';
117 1312 1 |
118 1313 1 |
119 1314 1 | MACROS:
120 1315 1 |
121 1316 1 |
122 1317 1 | EQUATED SYMBOLS:
123 1318 1 |
124 1319 1 | ! all other FORTRAN data type codes are larger valued
125 1320 1 |
126 1321 1 | PSECT DECLARATIONS:
127 1322 1 |
128 1323 1 DECLARE_PSECTS (FOR); ! declare PSECTS for FOR$ facility
129 1324 1 |
130 1325 1 | OWN STORAGE:
131 1326 1 |
132 1327 1 |     None
133 1328 1 |
134 1329 1 | EXTERNAL REFERENCES:
135 1330 1 |
136 1331 1 |
```

FORSSUDF\_WU  
1-012

D 2  
16-Sep-1984 00:55:07  
14-Sep-1984 12:32:57

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

Page 4  
(2)

```
137 1332 1 EXTERNAL
138 1333 1   FORSSAA_REC_PRO : VECTOR,
139 1334 1
140 1335 1
141 1336 1
142 1337 1   FORSSAA_REC_PR1 : VECTOR,
143 1338 1
144 1339 1
145 1340 1
146 1341 1   FORSSAA_REC_PR9 : VECTOR;
147 1342 1
148 1343 1
149 1344 1
150 1345 1
151 1346 1
152 1347 1 EXTERNAL ROUTINE
153 1348 1   FORSSIGNAL_STO : NOVALUE;
```

```
: PIC array of record processor
: procedure-initializations in REC
: level of abstraction. Indexed by
: I/O statement type (ISBSB_STM_TYPE)
: PIC array of record processor procedures
: Write a record in REC level of
: abstraction. Indexed by I/O statement
: type (ISBSB_STM_TYPE)
: PIC array of record processor procedures
:
: Write last record in REC level of
: abstraction. Indexed by I/O
: statement type (ISBSB_STM_TYPE)
:
: Signal fatal error
```

```

155 1349 1 GLOBAL ROUTINE FOR$$UDF_WUO ! Write unformatted UDF initialization
156 1350 1 : JSB_UDFO NOVALUE =
157 1351 1
158 1352 1 ++
159 1353 1 FUNCTIONAL DESCRIPTION:
160 1354 1
161 1355 1 Initialize Write unformatted User data formatter (UDF)
162 1356 1
163 1357 1 CALLING SEQUENCE:
164 1358 1
165 1359 1 JSB FOR$$UDF_WUO ( )
166 1360 1
167 1361 1 FORMAL PARAMETERS:
168 1362 1
169 1363 1 None.
170 1364 1
171 1365 1 IMPLICIT INPUTS:
172 1366 1
173 1367 1 CCB Pointer to current logical unit block
174 1368 1 ISB$B_STM_TYPE I/O statement type code - set by
175 1369 1 each I/O statement initialization
176 1370 1
177 1371 1 IMPLICIT OUTPUTS:
178 1372 1
179 1373 1 LUB$A_BUF_PTR Adr. of next byte of output
180 1374 1 data buffer
181 1375 1 LUB$A_BUF_END Adr. of end+1 byte position of output buffer
182 1376 1
183 1377 1 ROUTINE VALUE:
184 1378 1 COMPLETION CODES:
185 1379 1
186 1380 1 NONE
187 1381 1
188 1382 1 SIDE EFFECTS:
189 1383 1
190 1384 1 NONE
191 1385 1
192 1386 1 --
193 1387 1
194 1388 2 BEGIN
195 1389 2
196 1390 2 EXTERNAL REGISTER
197 1391 2 CCB : REF BLOCK [, BYTE];
198 1392 2
199 1393 2 ++
200 1394 2 Initialize Record processing level of abstraction.
201 1395 2 Set pointer to current (LUB$A_BUF_PTR) and last+1
202 1396 2 (LUB$A_BUF_END) character position for user data in
203 1397 2 output buffer
204 1398 2
205 1399 2
206 1400 2 JSB_RECO (FOR$$AA_REC_PRO + .FOR$$AA_REC_PRO [CCB [ISB$B_STM_TYPE] -
207 1401 2 ISB$K_FORSTTYLO + 1]);
208 1402 2
209 1403 2 RETURN;
210 1404 1 END; ! End of FOR$$UDF_WUO routine

```

FORSSUDF\_WU  
1-012

F 2  
16-Sep-1984 00:55:07 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:32:57 [FORRTL.SRC]FORUDFWU.B32;1

Page 6  
(3)

.TITLE FORSSUDF\_WU  
.IDENT \1-012\

.EXTRN FORSSAA\_REC\_PRO  
.EXTRN FORSSAA\_REC\_PR1  
.EXTRN FORSSAA\_REC\_PR9  
.EXTRN FORSSIGNAL\_STO

.PSECT \_FOR\$CODE,NOWRT, SHR, PIC,2

50 FF71 CB 9A 0000 FORSSUDF WU0::  
MOVZBL  
50 00000000G0040 D0 00005  
MOV L FORSSAA\_REC\_PRO[R0], R0  
00000000G0040 17 0000D  
JMP FORSSAA\_REC\_PRO[R0]

-143(CCB), R0  
FORSSAA\_REC\_PRO[R0], R0  
FORSSAA\_REC\_PRO[R0]

: 1401  
: 1400  
:

: Routine Size: 20 bytes, Routine Base: \_FOR\$CODE + 0000

: 211 1405 1





```
270 1463 1 | The following ISB locations are set only by previous calls
271 1464 1 | to FOR$$UDF_WU(0,1), i.e., are effectively OWN.
272 1465 1 |
273 1466 1 |     LUBSA_BUF_PTR           Pointer to next char. position
274 1467 1 |                             in user data part of output buffer
275 1468 1 |
276 1469 1 | FUNCTIONAL VALUE:
277 1470 1 |
278 1471 1 |     NONE
279 1472 1 |
280 1473 1 | SIDE EFFECTS:
281 1474 1 |
282 1475 1 |     SIGNAL_STOPs FOR$_OUTSTAOVE (Output statement overflows record)
283 1476 1 |
284 1477 1 |
285 1478 1 | BEGIN
286 1479 1 |
287 1480 1 | EXTERNAL REGISTER
288 1481 1 |     CCB : REF BLOCK [, BYTE];
289 1482 1 |
290 1483 1 |
291 1484 1 | +
292 1485 1 | | If this is a single-element list, check to see if the conditions
293 1486 1 | | for an unbuffered transfer are met.  If so, set RAB$L_RBF and
294 1487 1 | | RAB$W_RSZ to point directly to the element, and return leaving
295 1488 1 | | ISB$V_SINGL_ELEM set as an indication to REC level not to change
296 1489 1 | | them.  Otherwise clear ISB$V_SINGL_ELEM and proceed normally.
297 1490 1 | |
298 1491 1 | |
299 1492 1 | | IF .CCB [ISB$V_SINGL_ELEM]
300 1493 1 | | THEN
301 1494 1 | |     IF NOT .CCB [LUB$V_SEGMENTED] AND
302 1495 1 | |         (IF .CCB [LUB$V_FIXED] THEN .ELEM_SIZE EQLU .CCB [LUB$W_RBUF_SIZE]
303 1496 1 | |         ELSE .ELEM_SIZE LSSU .CCB [LUB$W_RBUF_SIZE])
304 1497 1 | |     THEN
305 1498 1 | |         BEGIN
306 1499 1 | |             CCB [RAB$L_RBF] = .ELEM_ADR;
307 1500 1 | |             CCB [RAB$W_RSZ] = .ELEM_SIZE;
308 1501 1 | |             RETURN;
309 1502 1 | |         END
310 1503 1 | |     ELSE
311 1504 1 | |         CCB [ISB$V_SINGL_ELEM] = 0;
312 1505 1 | |
313 1506 1 | | BEGIN
314 1507 1 | | LOCAL
315 1508 1 | |     TMP_ELEM_SIZE,           ! temp no. of bytes left in user element to copy
316 1509 1 | |     TMP_ELEM_ADR,           ! temp adr. of rest of user element to copy
317 1510 1 | |     TMP_DIFF;               ! temp. no. of bytes to move each time in loop
318 1511 1 | |
319 1512 1 | | +
320 1513 1 | | | If the record will overflow, write as much as will fit.  If this
321 1514 1 | | | is done, and the file is not SEGMENTED, then error FOR$_OUTSTAOVE.
322 1515 1 | | | If segmented, continue writing records until overflow is removed.
323 1516 1 | | |
324 1517 1 | | |
325 1518 1 | | | TMP_ELEM_SIZE = .ELEM_SIZE;
326 1519 1 | | | TMP_ELEM_ADR = .ELEM_ADR;
```



FORSSUDF\_WU  
1-012

J 2  
16-Sep-1984 00:55:07  
14-Sep-1984 12:32:57

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

Page 10  
(4)

			7E	42	8F	9A	00074		
		00000000G	00		01	FB	00078		
					C2	11	0007F		
B0	BB		67		58	28	00081	6\$:	
		B0	AB		53	D0	00086		
						04	0008A		

MOVZBL	#66, -(SP)	
CALLS	#1, FOR\$\$SIGNAL_STO	
BRB	5\$	
MOVCS	TMP_ELEM_SIZE, (TMP_ELEM_ADR), @-80(CCB)	
MOVL	R3, -80(CCB)	
RET		

:	1532
:	
:	1521
:	1540
:	
:	1543

: Routine Size: 139 bytes, Routine Base: \_FOR\$CODE + 0014

: 351 1544 1

```

: 353      1545 1 GLOBAL ROUTINE FOR$$UDF_WU9          ! unformatted output - end of I/O list call
: 354      1546 1   : JSB_UDF9 NOVALUE =
: 355      1547 1
: 356      1548 1
: 357      1549 1 +-
: 358      1550 1 FUNCTIONAL DESCRIPTION:
: 359      1551 1     FOR$$UDF_WU9 performs end of I/O list output formatting.
: 360      1552 1
: 361      1553 1 CALLING SEQUENCE:
: 362      1554 1
: 363      1555 1     JSB FOR$$UDF_WU9 ( )
: 364      1556 1
: 365      1557 1 FORMAL PARAMETERS:
: 366      1558 1
: 367      1559 1     NONE
: 368      1560 1
: 369      1561 1 IMPLICIT INPUTS:
: 370      1562 1
: 371      1563 1     See FOR$$UDF_WU1
: 372      1564 1
: 373      1565 1
: 374      1566 1 IMPLICIT OUTPUTS:
: 375      1567 1
: 376      1568 1     See FOR$$UDF_WU1
: 377      1569 1
: 378      1570 1 FUNCTION VALUE:
: 379      1571 1
: 380      1572 1     NONE
: 381      1573 1
: 382      1574 1 SIDE EFFECTS:
: 383      1575 1
: 384      1576 1     See FOR$$UDF_WU1
: 385      1577 1 --
: 386      1578 1 BEGIN
: 387      1579 2
: 388      1580 2 EXTERNAL REGISTER
: 389      1581 2 CCB : REF BLOCK [, BYTE];
: 390      1582 2
: 391      1583 2
: 392      1584 2 +-
: 393      1585 2 ! Call record level of abstraction to output buffer from beginning up to but not including LUB$A_BUF_PTR
: 394      1586 2 !-
: 395      1587 2
: 396      1588 2 JSB_REC9 (FOR$$AA_REC_PR9 + .FOR$$AA_REC_PR9 [,CCB [ISB$B_STM_TYPE] -
: 397      1589 2     ISB$K_FORSTTYLO + 1]);
: 398      1590 2
: 399      1591 2 RETURN;
: 400      1592 1 END;          ! End of FOR$$UDF_WU9 Routine

```

```

50      FF71  CB  9A 0000 FOR$$UDF_WU9::
50      00000000G0040  D0 00005      MOVZBL  -143(CCB), R0          : 1589
          00000000G0040  17 0000D      MOVL   FOR$$AA_REC_PR9[R0], R0      : 1588
          :
          :
          :

```

FOR\$\$UDF\_WU  
1-012

L 2  
16-Sep-1984 00:55:07  
14-Sep-1984 12:32:57

VAX-11 BlISS-32 V4.0-742  
[FORRTL.SRC]FORUDFWU.B32;1

Page 12  
(5)

: Routine Size: 20 bytes, Routine Base: \_FOR\$CODE + 009F

: 401 1593 1  
: 402 1594 1 END  
: 403 1595 1  
: 404 1596 0 ELUDOM

. End of FOR\$\$UDF\_WU Module

PSECT SUMMARY

Name	Bytes	Attributes
_FOR\$CODE	179	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	2 0	581	00:01.1

COMMAND QUALIFIERS

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

: Size: 179 code + 0 data bytes  
: Run Time: 00:13.3  
: Elapsed Time: 00:40.0  
: Lines/CPU Min: 7194  
: Lexemes/CPU-Min: 37523  
: Memory Used: 159 pages  
: Compilation Complete

