K 6

```
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**FORUDFWN                          E 16




FFFFFFFFFF    OOOOOO    RRRRRRRR    UU      UU  DDDDDDDD    FFFFFFFFFF  WW      WW  NN      NN
FFFFFFFFFF    OOOOOO    RRRRRRRR    UU      UU  DDDDDDDD    FFFFFFFFFF  WW      WW  NN      NN
FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  NN      NN
FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  NN      NN
FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  NNNN    NN
FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  NNNN    NN
FFFFFFFF    OO      OO  RRRRRRRR    UU      UU  DD      DD  FFFFFFFF    WW      WW  NN  NN  NN
FFFFFFFF    OO      OO  RRRRRRRR    UU      UU  DD      DD  FFFFFFFF    WW      WW  NN  NN  NN
FF          OO      OO  RR  RR      UU      UU  DD      DD  FF          WW  WW  WW  NN    NNNN
FF          OO      OO  RR   RR     UU      UU  DD      DD  FF          WW  WW  WW  NN    NNNN
FF          OO      OO  RR    RR    UU      UU  DD      DD  FF          WWWW  WWWW  NN      NN    ....
FF          OO      OO  RR    RR    UU      UU  DD      DD  FF          WWWW  WWWW  NN      NN    ....
FF            OOOOOO    RR      RR  UUUUUUUUUU  DDDDDDDD    FF          WW      WW  NN      NN    ....
FF            OOOOOO    RR      RR  UUUUUUUUUU  DDDDDDDD    FF          WW      WW  NN      NN    ....


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

```
0 MODULE FOR$$UDF_WN (%TITLE 'FOR$$UDF_WN - FORTRAN WRITE NAMELIST UDF Level'
0                IDENT = '1-005'           ! File: FORUDFWN.B32 Edit: SBL1005
0                ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     FORTRAN Compiled Code Support
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the User Data Formatter routines to perform
1 !       FORTRAN NAMELIST WRITE statements.
1 !
1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
1 !
1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 29-August-1980
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  SBL 29-August-1980
1 ! 1-002 - Reflect group block spec change where count-of-variables is a word;
1 !         second word is reserved.  SBL 5-Dec-1980
1 ! 1-003 - Add text describing NAMELIST descriptor block.  SBL 15-April-1981
1 ! 1-004 - REQUIRE FORERR.B32 instead of external reference to FOR$K_ symbols.
1 !         SBL 12-Aug-1981
1 ! 1-005 - Add ability to dump just the variable names for "?" feature.
1 !         Use prologue file.  SBL 24-May-1983
1 !--
1
```

```
1 %SBTTL 'Declarations'
1 !
1 ! PROLOGUE FILE:
1 !
1
1 REQUIRE 'RTLIN:FORPROLOG';                                   ! FOR$ definitions
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     FOR$$UDF_WN0: JSB_UDF0 NOVALUE,                          ! Start WRITE NAMELIST
1     FOR$$DO_NML_OUTPUT: CALL_CCB NOVALUE,                    ! Do bulk of processing
1     FOR$$UDF_WN9: JSB_UDF9 NOVALUE,                          ! End WRITE NAMELIST
1     CHECK_FIELD: CALL_CCB;                                   ! Check field width
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !       NONE
1 !
1 ! FIELDS:
1 !
1 !       NONE
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 EXTERNAL ROUTINE
1     FOR$$REC_WSN0: JSB_REC0 NOVALUE,                         ! Set up for a write
1     FOR$$REC_WSN1: JSB_REC1 NOVALUE,                         ! Write a record
1     FOR$$UDF_WL1: CALL_CCB NOVALUE,                          ! Convert and move to buffer
1     FOR$$SIGNAL_STO: NOVALUE;                                ! Signal fatal error
```

```
1 %SBTTL 'FOR$$UDF_WNO - Start WRITE NAMELIST'
1 GLOBAL ROUTINE FOR$$UDF_WNO: JSB_UDFO NOVALUE          ! Start WRITE NAMELIST
1     =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine starts a FORTRAN WRITE NAMELIST.  It calls FOR$$DO_NML_OUTPUT
1 !       to do the bulk of the work.  There is no UDF1 routine in this module
1 !       because WRITE NAMELIST statements have no I/O lists.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       JSB FOR$$UDF_WNO
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       CCB                             ! Register pointer to RAB/LUB/ISB
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       See FOR$$DO_NML_OUTPUT
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       See FOR$$DO_NML_OUTPUT
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         CCB = 11: REF $FOR$CCB_DECL;
2
2     FOR$$REC_WSNO ();
2
2     FOR$$DO_NML_OUTPUT (0);     ! Indicate that both names and values should print
2
2     RETURN;
2
1     END;                                            ! End of routine

                                        .TITLE  FOR$$UDF_WN FOR$$UDF_WN - FORTRAN WRITE NAMELIS
;                                                           T UDF Level
                                        .IDENT  \1-005\

                                        .EXTRN  FOR$$REC_WSNO, FOR$$REC_WSN1
                                        .EXTRN  FOR$$UDF_WL1, FOR$$SIGNAL_STO
```

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

                                   00000000G  00  16 00000 FOR$$UDF_WNO::
                                                                        JSB     FOR$$REC_WSNO                                     ; 0203
                                               7E  D4 00006             CLRL    -(SP)                                             ; 0205
                     0000V  CF                 01  FB 00008             CALLS   #1, FOR$$DO_NML_OUTPUT                            ;
                                                   05 0000D             RSB                                                       ; 0209

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

```
%SBTTL 'FOR$$DO_NML_OUTPUT - Do WRITE NAMELIST'
GLOBAL ROUTINE FOR$$DO_NML_OUTPUT (
    NAMES_ONLY                          ! Set if only names are to be printed
    ): CALL_CCB NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine performs one WRITE NAMELIST statement.
!
! CALLING SEQUENCE:
!
!       CALL FOR$$DO_NML_OUTPUT (NAMES_ONLY.rl.v)
!
! FORMAL PARAMETERS:
!
!       NAMES_ONLY                      0 if both names and values are to be printed
!                                       1 if only names are to be printed.  This can
!                                       occur if called from FOR$$UDF_RN to satisfy
!                                       a "?" inquiry.
!
! IMPLICIT INPUTS:
!
!       CCB                                     ! Register pointer to RAB/LUB/ISB
!       ISB$A_FMT_BEG                           ! Address of NAMELIST group descriptor
!
! IMPLICIT OUTPUTS:
!
!
! COMPLETION STATUS:
!
!       NONE
!
! SIDE EFFECTS:
!
!
!--

!<BLF/PAGE>
```

```
1 !++
1 !     Each NAMELIST descriptor block has the following form:
1 !
1 !            3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
1 !            1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
1 !
1 !           +---------------------------------------------------------------+
1 !         0 !              Address of ASCIC name of NAMELIST group          !
1 !           +---------------------------------------------------------------+
1 !         1 !            Reserved            !   Count of NAMELIST variables !
1 !           +---------------------------------------------------------------+
1 !         2 !               Address of ASCIC name of variable 1             !
1 !           +---------------------------------------------------------------+
1 !         3 !        Address of standard VAX descriptor for variable 1      !
1 !           +---------------------------------------------------------------+
1 !         4 !                             ...                               !
1 !           +---------------------------------------------------------------+
1 !         5 !               Address of ASCIC name of variable n             !
1 !           +---------------------------------------------------------------+
1 !         6 !        Address of standard VAX descriptor for variable n      !
1 !           +---------------------------------------------------------------+
1 !
1 !
1 !     The NAMELIST group name and the variable names which are pointed to in
1 !     the  NAMELIST  descriptor  block  are  upper  case  only.  The FORTRAN
1 !     compiler or other calling program is responsible for  case  conversion
1 !     of the name strings.  In NAMELIST input data, case is significant only
1 !     in character literals.  The run-time library is responsible  for  case
1 !     conversion of NAMELIST input data.
1 !
1 !     The allowable data types in variable descriptors are  BU  (BYTE),  WU,
1 !     LU,  W,  L,  F,  D, G, H, T, FC, DC, and GC.  The allowable descriptor
1 !     classes are scalar and array.  For the  array  class  descriptor,  the
1 !     descriptor  flags  COLUMN,  COEFF,  and BOUNDS must be set, indicating
1 !     column-major order and the presence of coefficient and bounds  blocks.
1 !     The number of dimensions must not exceed 7.
1 !--
1
1 !<BLF/PAGE>
2       BEGIN
2
2       EXTERNAL REGISTER
2           CCB = 11: REF $FOR$CCB_DECL;
2
2       LOCAL
2           GROUP: REF VECTOR [, LONG],                 ! NAMELIST group descriptor
2           NVARS,                                      ! Number of variables in group
2           VALUE_ADDR: REF VECTOR [, BYTE];            ! Address of value
2
2       !+
2       ! Write out group name
2       !-
2
2       GROUP = .CCB [ISB$A_FMT_BEG];
2       VALUE_ADDR = .GROUP [0];                        ! Address of group name counted string
2       IF NOT CHECK_FIELD (.VALUE_ADDR [0] + 2)        ! Include leading " $"
2       THEN
```

```
3           BEGIN
3           FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
3           RETURN;
2           END;
2       CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);     ! Write leading space
2       CH$WCHAR_A (%C'$', CCB [LUB$A_BUF_PTR]);     ! Write leading $
2       CCB [LUB$A_BUF_PTR] = CH$MOVE (.VALUE_ADDR [0], VALUE_ADDR [1], .CCB [LUB$A_BUF_PTR]);
2       FOR$$REC_WSN1 ();
2
2       !+
2       ! Scan through NAMELIST group and write all variables to the output stream.
2       !-
2
2
2       DECR NVARS FROM (.(GROUP [1])<0,16,0> - 1) TO 0 DO
3           BEGIN
3
3           LOCAL
3               OUT_NAME_LEN;                       ! Output name length
3
3           GROUP = GROUP [2];                      ! Skip to next variable
3           VALUE_ADDR = .GROUP [0];                ! Address of variable name counted string
3
3           !+
3           ! Compute output name length so that the names are padded to lengths
3           ! of 7, 15, 23, etc.  Add a space for before the "=".
3           !-
3
3           OUT_NAME_LEN = .VALUE_ADDR [0];
3           IF NOT .NAMES_ONLY
3           THEN
3               OUT_NAME_LEN = .OUT_NAME_LEN + (8 - (.VALUE_ADDR [0] MOD 8));
3
3           IF NOT CHECK_FIELD (.OUT_NAME_LEN + 2)  ! Include leading space, trailing " ="
3           THEN
4               BEGIN
4               FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
4               RETURN;
3               END;
3
3           !+
3           ! Write out variable name
3           !-
3
3           CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
3           CCB [LUB$A_BUF_PTR] = CH$COPY (.VALUE_ADDR [0], VALUE_ADDR [1],
3               %C' ', .OUT_NAME_LEN, .CCB [LUB$A_BUF_PTR]);
3
3           !+
3           ! Only print values if NAMES_ONLY is false.
3           !-
3
3           IF NOT .NAMES_ONLY
3           THEN
4               BEGIN
4               CH$WCHAR_A (%C'=', CCB [LUB$A_BUF_PTR]);
4
```

```
4                   !+
4                   ! Output all values in variable
4                   !-
4
5                       BEGIN
5                       LOCAL
5                           VAR_DESC: REF BLOCK [, BYTE],       ! Variable descriptor
5                           CUR_ADR,                    ! Current variable address
5                           END_ADR,                    ! End of variable
5                           ELEM_TYPE,                  ! Element datatype passed to FOR$$UDF_WL1
5                           CMPLX_FLAG;                 ! Complex flag passed to FOR$$UDF_WL1
5                           VAR_DESC = .GROUP [1];              ! Get descriptor
5                           CUR_ADR = .VAR_DESC [DSC$A_POINTER];
5                           IF .VAR_DESC [DSC$B_CLASS] EQL DSC$K_CLASS_A
5                           THEN
5                               END_ADR = .CUR_ADR + .VAR_DESC [DSC$L_ARSIZE]
5                           ELSE
5                               END_ADR = .CUR_ADR + .VAR_DESC [DSC$W_LENGTH];
5                           SELECTONE .VAR_DESC [DSC$B_DTYPE] OF
5                               SET
5                               [DSC$K_DTYPE_FC]:
6                                   BEGIN
6                                   ELEM_TYPE = DSC$K_DTYPE_F;
6                                   CMPLX_FLAG = 0;
5                                   END;
5                               [DSC$K_DTYPE_DC]:
6                                   BEGIN
6                                   ELEM_TYPE = DSC$K_DTYPE_D;
6                                   CMPLX_FLAG = 0;
5                                   END;
5                               [DSC$K_DTYPE_GC]:
6                                   BEGIN
6                                   ELEM_TYPE = DSC$K_DTYPE_G;
6                                   CMPLX_FLAG = 0;
5                                   END;
5                               [OTHERWISE]:
6                                   BEGIN
6                                   ELEM_TYPE = .VAR_DESC [DSC$B_DTYPE];
6                                   !+
6                                   ! FORTRAN passes us BU for B, so change it here.
6                                   IF .ELEM_TYPE EQL DSC$K_DTYPE_BU
6                                   THEN
6                                       ELEM_TYPE = DSC$K_DTYPE_B;
6                                   CMPLX_FLAG = 2;
5                                   END;
5                               TES;
5
5                       WHILE .END_ADR GTRA .CUR_ADR DO
6                           BEGIN
6                           LOCAL
6                               CUR_POS,
6                               REPEAT_COUNT;
6                           !+
6                           ! Build repeat count
6                           !-
6
6                           REPEAT_COUNT = 1;
```

```
CUR_POS = .CUR_ADR + .VAR_DESC [DSC$W_LENGTH];
WHILE .CUR_POS LSSA .END_ADR DO
    BEGIN
    IF NOT CH$EQL (.VAR_DESC [DSC$W_LENGTH],
                   .CUR_ADR,
                   .VAR_DESC [DSC$W_LENGTH],
                   .CUR_POS,
                   0)
    THEN
        EXITLOOP;
    CUR_POS = .CUR_POS + .VAR_DESC [DSC$W_LENGTH];
    REPEAT_COUNT = .REPEAT_COUNT + 1;
    END;

!+
! Is this variable of type CHARACTER?  If so, do all the
! processing here.  Otherwise, let FOR$$UDF_WL1 do most of
! the work.
!-

IF .ELEM_TYPE EQL DSC$K_DTYPE_T
THEN
    BEGIN
    !+
    ! It's CHARACTER.
    !-

    LOCAL
        REPEAT_DSC: DSC$DESCRIPTOR, ! Repeat string descriptor
        REPEAT_STR: VECTOR [12, BYTE],      ! Repeat string
        FAO_DSC: DSC$DESCRIPTOR;     ! FAO control string descr
    !+
    ! Build repeat count string.
    !-

    REPEAT_DSC [DSC$W_LENGTH] = 0;
    IF .REPEAT_COUNT GTR 1
    THEN
        BEGIN
        REPEAT_DSC [DSC$W_LENGTH] = 12;
        REPEAT_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
        REPEAT_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
        REPEAT_DSC [DSC$A_POINTER] = REPEAT_STR;
        FAO_DSC [DSC$A_POINTER] = UPLIT BYTE ('!SL*');
        FAO_DSC [DSC$W_LENGTH]  = %CHARCOUNT ('!SL*');
        FAO_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
        FAO_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
        $FAO (FAO_DSC,                  ! Control string
              REPEAT_DSC [DSC$W_LENGTH],     ! Returned length
              REPEAT_DSC,   ! Output string
              .REPEAT_COUNT);
        END;

    !+
    ! See if there is enough room for the repeat count
    !-
```

```
IF NOT CHECK_FIELD (2 + .REPEAT_DSC [DSC$W_LENGTH])
THEN
    BEGIN
    FOR$$REC_WSN1 ();
    IF NOT CHECK_FIELD (2 + .REPEAT_DSC [DSC$W_LENGTH])
    THEN
        FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
    END;

!+
! Write out a leading space, the repeat count and an
! initial apostrophe.
!-

CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
CCB [LUB$A_BUF_PTR] = CH$MOVE (.REPEAT_DSC [DSC$W_LENGTH], REPEAT_STR,
                                       .CCB [LUB$A_BUF_PTR]);
CH$WCHAR_A (%C'''', CCB [LUB$A_BUF_PTR]);

!+
! Write out each character of the string, substituting two
! apostrophes for each apostrophe found in the string.
!-

INCR I FROM 1 TO .VAR_DESC [DSC$W_LENGTH] DO
    BEGIN
    IF NOT CHECK_FIELD (1)
    THEN
        BEGIN
        FOR$$REC_WSN1 ();
        CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
        END;
    IF CH$RCHAR (.CUR_ADR) EQL %C''''
    THEN
        BEGIN
        CH$WCHAR_A (%C'''', CCB [LUB$A_BUF_PTR]);
        IF NOT CHECK_FIELD (1)
        THEN
            BEGIN
            FOR$$REC_WSN1 ();
            CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
            END;
        END;
    COPY_BYTE_A (CUR_ADR, CCB [LUB$A_BUF_PTR]);
    END;

!+
! Write out the closing apostrophe.
!-

IF NOT CHECK_FIELD (1)
THEN
    BEGIN
    FOR$$REC_WSN1 ();
    CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
    END;
CH$WCHAR_A (%C'''', CCB [LUB$A_BUF_PTR]);
```

```
7                               CUR_ADR = .CUR_POS;
7                               END
7
6                       ELSE
6
7                           BEGIN
7                           !+
7                           ! Not CHARACTER.
7                           ! Call list directed routine to output value.
7                           !-
7
7                               FOR$$UDF_WL1 (.ELEM_TYPE, .VAR_DESC [DSC$W_LENGTH], .CUR_ADR,
7                                               .CMPLX_FLAG, .REPEAT_COUNT);
7                           CUR_ADR = .CUR_POS;
6                           END;
6
6                       !+
6                       ! Put out a separating comma if values to come
6                       !-
6
6                       IF .CUR_ADR LSSA .END_ADR
6                       THEN
7                           BEGIN
7                           IF NOT CHECK_FIELD (1)
7                           THEN
8                               BEGIN
8                               FOR$$REC_WSN1 ();
8                               CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
7                               END;
7                           CH$WCHAR_A (%C',', CCB [LUB$A_BUF_PTR]);
6                           END;
5                       END;
4                   END;
4
4               !+
4               ! If this is not the last variable, write out a comma
4               !-
4
4               IF .NVARS NEQ 0
4               THEN
5                   BEGIN
5                   IF NOT CHECK_FIELD (1)
5                   THEN
6                       BEGIN
6                       FOR$$REC_WSN1 ();
6                       CH$WCHAR_A (%C' ', CCB [LUB$A_BUF_PTR]);
5                       END;
5                   CH$WCHAR_A (%C',', CCB [LUB$A_BUF_PTR]);
4                   END;
4
3               END;
3
3           !+
3           ! Write this record.
3           !-
3
3           FOR$$REC_WSN1 ();
```

```
;  530            0591  3
;  531            0592  2          END;
;  532            0593  2
;  533            0594  2      !+
;  534            0595  2      ! All variables written.  Put out $END block delimiter.
;  535            0596  2      !-
;  536            0597  2
;  537            0598  2      IF NOT CHECK_FIELD (5)
;  538            0599  2      THEN
;  539            0600  3          BEGIN
;  540            0601  3          FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
;  541            0602  3          RETURN;
;  542            0603  2          END;
;  543            0604  2      CCB [LUB$A_BUF_PTR] = CH$MOVE (5, UPLIT BYTE (' $END'), .CCB [LUB$A_BUF_PTR]);
;  544            0605  2      FOR$$REC_WSN1 ();
;  545            0606  2
;  546            0607  2      RETURN;
;  547            0608  2
;  548            0609  1      END;                                          ! End of routine



                                  2A  4C  53  21  0000E P.AAA:   .ASCII  \!SL*\                                     ;
                              44  4E  45  24  20  00012 P.AAB:   .ASCII  \ $END\                                    ;

                                                                 .EXTRN  SYS$FAO

                                         07FC 00000              .ENTRY  FOR$$DO_NML_OUTPUT, Save R2,R3,R4,R5,R6,R7,-; 0211
                                                                         R8,R9,R10                                  ;
                              5E          34  C2 00002           SUBL2   #52, SP                                    ;
                                   FF7C   CB  DD 00005           PUSHL   -132(CCB)                                  ; 0302
                              5A     00   BE  D0 00009           MOVL    @GROUP, VALUE_ADDR                         ; 0303
                              7E          6A  9A 0000D           MOVZBL  (VALUE_ADDR), -(SP)                        ; 0304
                              6E          02  C0 00010           ADDL2   #2, (SP)                                   ;
                      0000V   CF          01  FB 00013           CALLS   #1, CHECK_FIELD                            ;
                              5D          50  E9 00018           BLBC    R0, 3$                                     ;
                              56     B0   AB  9E 0001B           MOVAB   -80(CCB), R6                               ; 0310
                        00    B6          20  90 0001F           MOVB    #32, @0(R6)                                ;
                                          66  D6 00023           INCL    (R6)                                       ;
                        00    B6          24  90 00025           MOVB    #36, @0(R6)                                ; 0311
                                          66  D6 00029           INCL    (R6)                                       ;
                              50          6A  9A 0002B           MOVZBL  (VALUE_ADDR), R0                           ; 0312
              00    B6  01    AA          50  28 0002E           MOVC3   R0, 1(VALUE_ADDR), @0(R6)                  ;
                              66          53  D0 00034           MOVL    R3, (R6)                                   ;
                                00000000G 00  16 00037           JSB     FOR$$REC_WSN1                              ; 0313
                        18    AE     04   AC  D2 0003D           MCOML   NAMES_ONLY, 24(SP)                         ; 0335
                  7E          6E          04  C1 00042           ADDL3   #4, GROUP, -(SP)                           ;
                        14    AE          9E  3C 00046           MOVZWL  @(SP)+, NVARS                              ;
                                     0231 31 0004A           BRW     32$                                        ;
                              6E          08  C0 0004D 1$:       ADDL2   #8, GROUP                                  ; 0326
                              5A     00   BE  D0 00050           MOVL    @GROUP, VALUE_ADDR                         ; 0327
                              52          6A  9A 00054           MOVZBL  (VALUE_ADDR), OUT_NAME_LEN                 ; 0334
                              15     18   AE  E9 00057           BLBC    24(SP), 2$                                 ; 0335
                              50          6A  9A 0005B           MOVZBL  (VALUE_ADDR), R0                           ; 0337
        7E        00          50          01  7A 0005E           EMUL    #1, R0, #0, -(SP)                          ;
        50        50          8E          08  7B 00063           EDIV    #8, (SP)+, R0, R0                          ;
                  50          52          50  C3 00068           SUBL3   R0, OUT_NAME_LEN, R0                       ;
```

```
                                                          F  1
FOR$$UDF_WN     FOR$$UDF_WN - FORTRAN WRITE NAMELIST UDF Level  16-Sep-1984 00:53:55    VAX-11 Bliss-32 V4.0-742             Page  13
1-005           FOR$$DO_NML_OUTPUT - Do WRITE NAMELIST          14-Sep-1984 12:32:56    [FORRTL.SRC]FORUDFWN.B32;1                 (5)

                            52    08    A0  9E 0006C        MOVAB   8(R0), OUT_NAME_LEN                             ;
                                  02    A2  9F 00070  2$:   PUSHAB  2(OUT_NAME_LEN)                                 ; 0339
                     0000V    CF        01  FB 00073        CALLS   #1, CHECK_FIELD                                 ;
                            03          50  E8 00078  3$:   BLBS    R0, 4$                                          ;
                                      0213  31 0007B        BRW     35$                                             ;
                            56    B0    AB  9E 0007E  4$:   MOVAB   -80(CCB), R6                                    ; 0350
                        00  B6          20  90 00082        MOVB    #32, @0(R6)                                     ;
                                        66  D6 00086        INCL    (R6)                                            ;
                            50          6A  9A 00088        MOVZBL  (VALUE_ADDR), R0                                ; 0351
    52            20    01  AA          50  2C 0008B        MOVC5   R0, 1(VALUE_ADDR), #32, OUT_NAME_LEN, -         ; 0352
                                  00    B6     00091                @0(R6)                                          ;
                            66          53  D0 00093        MOVL    R3, (R6)                                        ;
                            03    18    AE  E8 00096        BLBS    24(SP), 5$                                      ; 0358
                                      01DB  31 0009A        BRW     31$                                             ;
                        00  B6          3D  90 0009D  5$:   MOVB    #61, @0(R6)                                     ; 0361
                                        66  D6 000A1        INCL    (R6)                                            ;
                  50        6E          04  C1 000A3        ADDL3   #4, GROUP, R0                                   ; 0374
                            57          60  D0 000A7        MOVL    (R0), VAR_DESC                                  ;
                            59    04    A7  D0 000AA        MOVL    4(VAR_DESC), CUR_ADR                            ; 0375
                            04    03    A7  91 000AE        CMPB    3(VAR_DESC), #4                                 ; 0376
                                        08  12 000B2        BNEQ    6$                                              ;
                        10  AE    0C  B749  9E 000B4        MOVAB   @12(VAR_DESC)[CUR_ADR], END_ADR                 ; 0378
                                        08  11 000BA        BRB     7$                                              ;
                            50          67  3C 000BC  6$:   MOVZWL  (VAR_DESC), R0                                  ; 0380
            10    AE        59          50  C1 000BF        ADDL3   R0, CUR_ADR, END_ADR                            ;
                            50    02    A7  9A 000C4  7$:   MOVZBL  2(VAR_DESC), R0                                 ; 0381
                            0C          50  91 000C8        CMPB    R0, #12                                         ; 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  8$:   CMPB    R0, #13                                         ; 0388
                                        05  12 000D5        BNEQ    9$                                              ;
                            58          0B  D0 000D7        MOVL    #11, ELEM_TYPE                                  ; 0390
                                        08  11 000DA        BRB     10$                                             ; 0391
                            1D          50  91 000DC  9$:   CMPB    R0, #29                                         ; 0393
                                        08  12 000DF        BNEQ    11$                                             ;
                            58          1B  D0 000E1        MOVL    #27, ELEM_TYPE                                  ; 0395
                                  0C    AE  D4 000E4  10$:  CLRL    CMPLX_FLAG                                      ; 0396
                                        0F  11 000E7        BRB     13$                                             ; 0381
                            58          50  D0 000E9  11$:  MOVL    R0, ELEM_TYPE                                   ; 0400
                            02          58  D1 000EC        CMPL    ELEM_TYPE, #2                                   ; 0403
                                        03  12 000EF        BNEQ    12$                                             ;
                            58          06  D0 000F1        MOVL    #6, ELEM_TYPE                                   ; 0405
                        0C  AE          02  D0 000F4  12$:  MOVL    #2, CMPLX_FLAG                                  ; 0406
                            59    10    AE  D1 000F8  13$:  CMPL    END_ADR, CUR_ADR                                ; 0410
                                        03  1A 000FC        BGTRU   14$                                             ;
                                      0154  31 000FE        BRW     29$                                             ;
                            54          01  D0 00101  14$:  MOVL    #1, REPEAT_COUNT                                ; 0419
                        04  AE          67  3C 00104        MOVZWL  (VAR_DESC), 4(SP)                               ; 0420
                        08  AE    04  BE49  9E 00108        MOVAB   @4(SP)[CUR_ADR], CUR_POS                        ;
                        10  AE    08    AE  D1 0010E  15$:  CMPL    CUR_POS, END_ADR                                ; 0421
                                        11  1E 00113        BGEQU   16$                                             ;
            08    BE        69    04    AE  29 00115        CMPC3   4(SP), (CUR_ADR), @CUR_POS                      ; 0423
                                        09  12 0011B        BNEQ    16$                                             ;
                        08  AE    04    AE  C0 0011D        ADDL2   4(SP), CUR_POS                                  ; 0430
                                        54  D6 00122        INCL    REPEAT_COUNT                                    ; 0431
                                        E8  11 00124        BRB     15$                                             ; 0421
```

```
                           0E            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                      ; 0459
                      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, SYS$FAO                                ;
                           52       30   AE  3C 00163 18$:     MOVZWL   REPEAT_DSC, R2                             ; 0477
                           52            02  C0 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      FOR$$REC_WSN1                              ; 0480
                                         52  DD 0017A          PUSHL    R2                                         ; 0481
                    0000V  CF            01  FB 0017C          CALLS    #1, CHECK_FIELD                            ;
                           0B            50  E8 00181          BLBS     R0, 19$                                    ;
                           7E       42   8F  9A 00184          MOVZBL   #66, -(SP)                                 ; 0483
                00000000G  00            01  FB 00188          CALLS    #1, FOR$$SIGNAL_STO                        ;
                           56       B0   AB  9E 0018F 19$:     MOVAB    -80(CCB), R6                               ; 0491
                      00   B6            20  90 00193          MOVB     #32, @0(R6)                                ;
                                         66  D6 00197          INCL     (R6)                                       ;
         00   B6      24   AE       30   AE  28 00199          MOVC3    REPEAT_DSC, REPEAT_STR, @0(R6)             ; 0493
                           66            53  D0 001A0          MOVL     R3, (R6)                                   ;
                      00   B6            27  90 001A3          MOVB     #39, @0(R6)                                ; 0494
                                         66  D6 001A7          INCL     (R6)                                       ;
                                         52  D4 001A9          CLRL     I                                          ; 0501
                                         45  11 001AB          BRB      23$                                        ;
                                         01  DD 001AD 20$:     PUSHL    #1                                         ; 0503
                    0000V  CF            01  FB 001AF          CALLS    #1, CHECK_FIELD                            ;
                           0D            50  E8 001B4          BLBS     R0, 21$                                    ;
                                00000000G 00  16 001B7          JSB      FOR$$REC_WSN1                              ; 0506
                      B0   BB            20  90 001BD          MOVB     #32, @-80(CCB)                             ; 0507
                                    B0   AB  D6 001C1          INCL     -80(CCB)                                   ;
                           27            69  91 001C4 21$:     CMPB     (CUR_ADR), #39                             ; 0509
                                         1E  12 001C7          BNEQ     22$                                        ;
                      B0   BB            27  90 001C9          MOVB     #39, @-80(CCB)                             ; 0512
                                    B0   AB  D6 001CD          INCL     -80(CCB)                                   ;
                                         01  DD 001D0          PUSHL    #1                                         ; 0513
                    0000V  CF            01  FB 001D2          CALLS    #1, CHECK_FIELD                            ;
                           0D            50  E8 001D7          BLBS     R0, 22$                                    ;
                                00000000G 00  16 001DA          JSB      FOR$$REC_WSN1                              ; 0516
                      B0   BB            20  90 001E0          MOVB     #32, @-80(CCB)                             ; 0517
                                    B0   AB  D6 001E4          INCL     -80(CCB)                                   ;
                                    B0   AB  D6 001E7 22$:     INCL     -80(CCB)                                   ; 0520
                           50       B0   AB  D0 001EA          MOVL     -80(CCB), R0                               ;
                      FF   A0            89  90 001EE          MOVB     (CUR_ADR)+, -1(R0)                         ;
              B6           52       04   AE  F3 001F2 23$:     AOBLEQ   4(SP), I, 20$                              ; 0501
                                         01  DD 001F7          PUSHL    #1                                         ; 0527
                    0000V  CF            01  FB 001F9          CALLS    #1, CHECK_FIELD                            ;
```

```
                             0D              50  E8 001FE          BLBS     R0, 24$                                   ;
                      00000000G  00          16  00201             JSB      FOR$$REC_WSN1                             ; 0530
                   B0    BB                  20  90 00207          MOVB     #32, @-80(CCB)                            ; 0531
                                   B0        AB  D6 0020B          INCL     -80(CCB)                                  ;
                   B0    BB                  27  90 0020E  24$:    MOVB     #39, @-80(CCB)                            ; 0533
                                   B0        AB  D6 00212          INCL     -80(CCB)                                  ;
                                             13  11 00215          BRB      26$                                       ; 0534
                                             54  DD 00217  25$:    PUSHL    REPEAT_COUNT                              ; 0546
                                   10        AE  DD 00219          PUSHL    CMPLX_FLAG                                ;
                                             59  DD 0021C          PUSHL    CUR_ADR                                   ; 0545
                                   10        AE  DD 0021E          PUSHL    16(SP)                                    ;
                                             58  DD 00221          PUSHL    ELEM_TYPE                                 ;
              00000000G  00                  05  FB 00223          CALLS    #5, FOR$$UDF_WL1                          ;
                         59        08        AE  D0 0022A  26$:    MOVL     CUR_POS, CUR_ADR                          ; 0547
                   10    AE                  59  D1 0022E          CMPL     CUR_ADR, END_ADR                          ; 0554
                                             1E  1E 00232          BGEQU    28$                                       ;
                                             01  DD 00234          PUSHL    #1                                        ; 0557
                 0000V   CF                  01  FB 00236          CALLS    #1, CHECK_FIELD                           ;
                         0D                  50  E8 0023B          BLBS     R0, 27$                                   ;
                      00000000G  00          16  0023E             JSB      FOR$$REC_WSN1                             ; 0560
                   B0    BB                  20  90 00244          MOVB     #32, @-80(CCB)                            ; 0561
                                   B0        AB  D6 00248          INCL     -80(CCB)                                  ;
                   B0    BB                  2C  90 0024B  27$:    MOVB     #44, @-80(CCB)                            ; 0563
                                   B0        AB  D6 0024F          INCL     -80(CCB)                                  ;
                                           FEA3  31 00252  28$:    BRW      13$                                       ; 0410
                                   14        AE  D5 00255  29$:    TSTL     NVARS                                     ; 0572
                                             1E  13 00258          BEQL     31$                                       ;
                                             01  DD 0025A          PUSHL    #1                                        ; 0575
                 0000V   CF                  01  FB 0025C          CALLS    #1, CHECK_FIELD                           ;
                         0D                  50  E8 00261          BLBS     R0, 30$                                   ;
                      00000000G  00          16  00264             JSB      FOR$$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  30$:    MOVB     #44, @-80(CCB)                            ; 0581
                                   B0        AB  D6 00275          INCL     -80(CCB)                                  ;
                      00000000G  00          16  00278     31$:    JSB      FOR$$REC_WSN1                             ; 0590
                         02        14        AE  F4 0027E  32$:    SOBGEQ   NVARS, 33$                                ; 0320
                                             03  11 00282          BRB      34$                                       ;
                                           FDC6  31 00284  33$:    BRW      1$                                        ;
                                             05  DD 00287  34$:    PUSHL    #5                                        ; 0598
                 0000V   CF                  01  FB 00289          CALLS    #1, CHECK_FIELD                           ;
                         0C                  50  E8 0028E          BLBS     R0, 36$                                   ;
                         7E        42        8F  9A 00291  35$:    MOVZBL   #66, -(SP)                                ; 0601
              00000000G  00                  01  FB 00295          CALLS    #1, FOR$$SIGNAL_STO                       ;
                                                 04 0029C          RET                                                ; 0600
          B0    BB     FD59  CF              05  28 0029D  36$:    MOVC3    #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 ! FUNCTIONAL DESCRIPTION:
;  557         0617  1 !
;  558         0618  1 !       End a namelist-directed WRITE statement.  This procedure, although
;  559         0619  1 !       a no-op, is necessary because FOR$IO_END dispatches to a UDF9 routine
;  560         0620  1 !       based on the statement type.
;  561         0621  1 !
;  562         0622  1 ! CALLING SEQUENCE:
;  563         0623  1 !
;  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 00000 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 00000 CHECK_FIELD:
                                                        .WORD   Save nothing                              ; 0655
              51      B0   AB      04   AC  C1 00002    ADDL3   WIDTH, -80(CCB), R1                       ; 0697
                                            50  D4 00008    CLRL    R0                                        ;
                      B4   AB           51  D1 0000A    CMPL    R1, -76(CCB)                              ;
                                            02  1A 0000E    BGTRU   1$                                        ;
```

```
                                          50  D6 00010            INCL    R0                                    ;
                                              04 00012  1$:       RET                                           ; 0699

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


;  641          0700  1 !<BLF/PAGE>
;  643          0701  1 END                                              ! End of module FOR$$UDF_WN
;  644          0702  1
;  645          0703  0 ELUDOM




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




;                             Library Statistics
;
;                                          -------- Symbols --------        Pages       Processing
;      File                                Total    Loaded   Percent        Mapped      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=LIS$:FORUDFWN/OBJ=OBJ$:FORUDFWN MSRC$:FORUDFWN/UPDATE=(ENH$: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
```

0184 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY

FORSTOP
LIS

FORUDFRL
LIS

FORTIMEDS
LIS

FORUDFWN
LIS

FORUDFWL
LIS

FORUDFRN
LIS

FORUDFRU
LIS

FORTIME
LIS

FORUDFWF
LIS

FORUDFRF
LIS

0185 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

HELP

FORWRITDF
LIS

VMSHELP
MAP

HLD
MDL

FORWRITDU
LIS

FORVECTOR
LIS

FORWRITIL
LIS

FORWRITSF
LIS

HLD

HLDMACROS
MAR

FORVM
LIS

FORWRITSN
LIS

HLD
MAP

FORWRITSU
LIS

RSXDEFS
MAR

FORUNLOCK
LIS

FORWRITDO
LIS

HELP
LIS

FORWRITIF
LIS

FORWRITIO
LIS

HLDCSTA
LIS

FORUDFWU
LIS

FORUNDERF
LIS

FORWRITSL
LIS

FORWRITSO
LIS