```
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFFFFFFFFFF       000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFFFFFFFFFF       000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFFFFFFFFFF       000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFF               000         000 RRR   RRR       RRR   RRR             TTT        LLL
FFF               000         000 RRR   RRR       RRR   RRR             TTT        LLL
FFF               000         000 RRR   RRR       RRR   RRR             TTT        LLL
FFF               000         000 RRR     RRR     RRR     RRR           TTT        LLL
FFF               000         000 RRR     RRR     RRR     RRR           TTT        LLL
FFF               000         000 RRR     RRR     RRR     RRR           TTT        LLL
FFF                 000000000     RRR         RRR RRR         RRR       TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR         RRR RRR         RRR       TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR         RRR RRR         RRR       TTT        LLLLLLLLLLLLLLL
```

FILEID**FORUNDERF

M 2

```
FFFFFFFFFF    000000    RRRRRRRR    UU      UU  NN      NN  DDDDDDDD    EEEEEEEEEE  RRRRRRRR    FFFFFFFFFF
FFFFFFFFFF    000000    RRRRRRRR    UU      UU  NN      NN  DDDDDDDD    EEEEEEEEEE  RRRRRRRR    FFFFFFFFFF
FF          00      00  RR      RR  UU      UU  NN      NN  DD      DD  EE          RR      RR  FF
FF          00      00  RR      RR  UU      UU  NN      NN  DD      DD  EE          RR      RR  FF
FF          00      00  RR      RR  UU      UU  NNNN    NN  DD      DD  EE          RR      RR  FF
FF          00      00  RR      RR  UU      UU  NNNN    NN  DD      DD  EE          RR      RR  FF
FFFFFFFF    00      00  RRRRRRRR    UU      UU  NN  NN  NN  DD      DD  EEEEEEEE    RRRRRRRR    FFFFFFFF
FFFFFFFF    00      00  RRRRRRRR    UU      UU  NN  NN  NN  DD      DD  EEEEEEEE    RRRRRRRR    FFFFFFFF
FF          00      00  RR  RR      UU      UU  NN    NNNN  DD      DD  EE          RR  RR      FF
FF          00      00  RR  RR      UU      UU  NN    NNNN  DD      DD  EE          RR  RR      FF
FF          00      00  RR    RR    UU      UU  NN      NN  DD      DD  EE          RR    RR    FF
FF          00      00  RR    RR    UU      UU  NN      NN  DD      DD  EE          RR    RR    FF
FF            000000    RR      RR  UUUUUUUUUU  NN      NN  DDDDDDDD    EEEEEEEEEE  RR      RR  FF
FF            000000    RR      RR  UUUUUUUUUU  NN      NN  DDDDDDDD    EEEEEEEEEE  RR      RR  FF

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
```

```
;  1          0001  0 %TITLE 'FOR$UNDERF - Fortran underflow exception handler'
;  2          0002  0 MODULE FOR$UNDERF (
;  3          0003  0                    IDENT = '1-003'          ! File: FORUNDERF.B32  Edit: JAW1003
;  4          0004  0                    ) =
;  5          0005  1 BEGIN
;  6          0006  1 !++
;  7          0007  1 !
;  8          0008  1 !****************************************************************************
;  9          0009  1 !*                                                                          *
; 10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
; 11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
; 12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
; 13          0013  1 !*                                                                        *
; 14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
; 17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 19          0019  1 !*  TRANSFERRED.                                                           *
; 20          0020  1 !*                                                                        *
; 21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 23          0023  1 !*  CORPORATION.                                                           *
; 24          0024  1 !*                                                                        *
; 25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
; 27          0027  1 !*                                                                        *
; 28          0028  1 !*                                                                        *
; 29          0029  1 !****************************************************************************
; 30          0030  1 !
; 31          0031  1 !
; 32          0032  1 ! FACILITY:     Fortran Support Library
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT:
; 35          0035  1 !
; 36          0036  1 !       This module contains a condition handler for floating underflow
; 37          0037  1 !       exceptions and an exit handler to report the number of underflow
; 38          0038  1 !       exceptions at image exit.
; 39          0039  1 !
; 40          0040  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
; 41          0041  1 !
; 42          0042  1 ! AUTHOR: John A. Wheeler, CREATION DATE: 21-Aug-1981
; 43          0043  1 !
; 44          0044  1 ! MODIFIED BY:
; 45          0045  1 !
; 46          0046  1 ! 1-001 - Original.  JAW 21-Aug-1981
; 47          0047  1 ! 1-002 - Remove address of UNDERFLOW_COUNT from exit control block, as
; 48          0048  1 !         count is now referenced directly.  Remove unused external
; 49          0049  1 !         reference to LIB$MATCH_COND.  JAW 25-Aug-1981
; 50          0050  1 ! 1-003 - Change name of FOR$HANDLER to FOR$UNDERFLOW_HANDLER.  Give the
; 51          0051  1 !         threshold value a non-public name.  Include severity in
; 52          0052  1 !         condition check to preclude counting the same exception twice
; 53          0053  1 !         at more than one level.  JAW 29-Aug-1981
; 54          0054  1 !--
; 55          0055  1
```

```
;    57          0056  1 %SBTTL 'Declarations'
;    58          0057  1 !
;    59          0058  1 ! SWITCHES:
;    60          0059  1 !
;    61          0060  1
;    62          0061  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    63          0062  1
;    64          0063  1 !
;    65          0064  1 ! LINKAGES:
;    66          0065  1 !
;    67          0066  1 !       NONE
;    68          0067  1 !
;    69          0068  1 ! TABLE OF CONTENTS:
;    70          0069  1 !
;    71          0070  1
;    72          0071  1 FORWARD ROUTINE
;    73          0072  1     FOR$UNDERFLOW_HANDLER,
;    74          0073  1     EXIT_HANDLER : NOVALUE;
;    75          0074  1
;    76          0075  1 !
;    77          0076  1 ! INCLUDE FILES:
;    78          0077  1 !
;    79          0078  1
;    80          0079  1 LIBRARY 'RTLSTARLE';
;    81          0080  1
;    82          0081  1 REQUIRE 'RTLML:FORMSG';
;    83          0246  1 REQUIRE 'RTLML:OTSMSG';
;    84          0333  1 REQUIRE 'RTLML:MTHMSG';
;    85          0409  1
;    86          0410  1 REQUIRE 'RTLIN:RTLPSECT';               ! Define PSECT declaration macros
;    87          0505  1
;    88          0506  1 !
;    89          0507  1 ! MACROS:
;    90          0508  1 !
;    91          0509  1 !       NONE
;    92          0510  1 !
;    93          0511  1 ! EQUATED SYMBOLS:
;    94          0512  1 !
;    95          0513  1
;    96          0514  1 LITERAL
;    97          0515  1     K_UNDERFLOW_THRESHOLD = 2;          ! Message-printing threshold
;    98          0516  1
;    99          0517  1 !
;   100          0518  1 ! FIELDS:
;   101          0519  1 !
;   102          0520  1 !       NONE
;   103          0521  1 !
;   104          0522  1 ! PSECTS:
;   105          0523  1 !
;   106          0524  1 !       Specify page alignment (9) for the OWN psect, so that EXIT_BLOCK
;   107          0525  1 !       will not occupy the same page as a user variable that is being
;   108          0526  1 !       WATCHed, and thus be unwriteable when $DCLEXH is called.
;   109          0527  1
;   110          0528  1 DECLARE_PSECTS (FOR, 9);               ! Declare PSECTs for FOR$ facility
;   111          0529  1
;   112          0530  1 !
;   113          0531  1 ! OWN STORAGE:
```

```
;  114          0532  1 !
;  115          0533  1 ;
;  116          0534  1     OWN
;  117          0535  1         UNDERFLOW_COUNT : VOLATILE,      ! Number of underflows which have
;  118          0536  1                                          !  occurred (and have reached FOR$UNDERFLOW_HANDLER)
;  119          0537  1         EXIT_HNDLR_LOCK : INITIAL (0)    ! Flag indicating whether exit
;  120          0538  1             VOLATILE;                    !  handler has been declared yet
;  121          0539  1
;  122          0540  1 !
;  123          0541  1 ! EXTERNAL REFERENCES:
;  124          0542  1 !
;  125          0543  1
;  126          0544  1 EXTERNAL ROUTINE
;  127          0545  1     LIB$SIM_TRAP,
;  128          0546  1     LIB$SIGNAL,
;  129          0547  1     LIB$STOP;
```

```
;  131          0548  1 GLOBAL ROUTINE FOR$UNDERFLOW_HANDLER (    ! FORTRAN floating underflow handler
;  132          0549  1 SIG_ARGS_ADR,                             ! Address of signal array
;  133          0550  1 MCH_ARGS_ADR,                             ! Address of mecharism array
;  134          0551  1 ENB_ARGS_ADR                              ! Address of enable array
;  135          0552  1     ) =                                   ! A handler always has a value
;  136          0553  1
;  137          0554  2     BEGIN
;  138          0555  2
;  139          0556  2     MAP
;  140          0557  2         SIG_ARGS_ADR : REF BLOCK [, BYTE],
;  141          0558  2         MCH_ARGS_ADR : REF BLOCK [, BYTE],
;  142          0559  2         ENB_ARGS_ADR : REF VECTOR;
;  143          0560  2
;  144          0561  2     OWN
;  145          0562  2         EXIT_REASON,                      ! Reason for image exit (not used)
;  146          0563  2
;  147          0564  2         EXIT_BLOCK : VECTOR [4] INITIAL ! Exit control block for $DCLEXH
;  148          0565  2             (0,                           !  Forward link (filled in by VMS)
;  149          0566  2              0,                           !  Exit handler address
;  150          0567  2              2,                           !  Number of arguments to exit handler
;  151          0568  2              0);                          .  Address of EXIT_REASON
;  152          0569  2
;  153          0570  2     LOCAL
;  154          0571  2         DCLEXH_STATUS,                    ! Result of $DCLEXH
;  155          0572  2         AST_STATUS;                       ! Result of $SETAST
;  156          0573  2
;  157          0574  2     !+
;  158          0575  2     ! If the exception is any form of floating underflow (fault, trap or
;  159          0576  2     ! math library condition), count it, print a message if not yet over
;  160          0577  2     ! the limit, and continue.
;  161          0578  2     !-
;  162          0579  2
;  163          0580  2     IF (.SIG_ARGS_ADR [CHF$L_SIG_NAME] AND (STS$M_COND_ID OR STS$M_SEVERITY)) EQL
;  164          0581  3                         (SS$_FLTUND_F AND (STS$M_COND_ID OR STS$M_SEVERITY))
;  165          0582  2     THEN
;  166          0583  2
;  167          0584  2         !+
;  168          0585  2         ! A fault.
;  169          0586  2         !
;  170          0587  2         ! Convert the fault to a trap.  Control will not return from
;  171          0588  2         ! LIB$SIM_TRAP; rather, a new exception (a trap) will occur.
;  172          0589  2         !-
;  173          0590  2
;  174          0591  2         LIB$SIM_TRAP (.SIG_ARGS_ADR, .MCH_ARGS_ADR)
;  175          0592  2
;  176          0593  2     ELSE
;  177          0594  3         IF ((.SIG_ARGS_ADR [CHF$L_SIG_NAME] AND (STS$M_COND_ID OR STS$M_SEVERITY)) EQL
;  178          0595  3                                 (SS$_FLTUND AND (STS$M_COND_ID OR STS$M_SEVERITY)))
;  179          0596  2             OR
;  180          0597  3            ((.SIG_ARGS_ADR [CHF$L_SIG_NAME] AND (STS$M_COND_ID OR STS$M_SEVERITY)) EQL
;  181          0598  3                               (MTH$_FLOONDMAT AND (STS$M_COND_ID OR STS$M_SEVERITY)))
;  182          0599  2         THEN
;  183          0600  3             BEGIN
;  184          0601  3
;  185          0602  3             !+
;  186          0603  3             ! A trap or math library condition.
;  187          0604  3             !
```

```
;  188          0605  3            ! Count the underflow.  Then check EXIT_HNDLR_LOCK to see
;  189          0606  3            ! whether the exit handler has been declared yet (1 = yes,
;  190          0607  3            ! 0 = maybe).  If maybe, disable ASTs and recheck.  This
;  191          0608  3            ! assures that the exit handler will be declared only once
;  192          0609  3            ! even if underflows occur at AST level.
;  193          0610  3            !-
;  194          0611  3
;  195          0612  3            UNDERFLOW_COUNT = .UNDERFLOW_COUNT + 1;
;  196          0613  3
;  197          0614  3            IF NOT .EXIT_HNDLR_LOCK
;  198          0615  3            THEN
;  199          0616  4                BEGIN
;  200          0617  4                AST_STATUS = $SETAST (ENBFLG = 0);
;  201          0618  4                IF NOT .EXIT_HNDLR_LOCK
;  202          0619  4                THEN
;  203          0620  5                    BEGIN
;  204          0621  5
;  205          0622  5                    !+
;  206          0623  5                    ! Fill in the exit control block (at run time, to
;  207          0624  5                    ! keep it position-independent), declare the exit
;  208          0625  5                    ! handler, and set the lock.
;  209          0626  5                    !-
;  210          0627  5
;  211          0628  5                    EXIT_BLOCK [1] = EXIT_HANDLER;
;  212          0629  5                    EXIT_BLOCK [3] = EXIT_REASON;
;  213          0630  5                    DCLEXH_STATUS = $DCLEXH (DESBLK = EXIT_BLOCK);
;  214          0631  5                    EXIT_HNDLR_LOCK = 1
;  215          0632  5                    END
;  216          0633  4                ELSE
;  217          0634  4                    DCLEXH_STATUS = 1;
;  218          0635  4
;  219          0636  4                IF .AST_STATUS EQL SS$_WASSET THEN $SETAST (ENBFLG = 1);
;  220          0637  4
;  221          0638  4                IF NOT .DCLEXH_STATUS THEN LIB$STOP (OTS$_FATINTERR) ELSE 1
;  222          0639  3                END;
;  223          0640  3
;  224          0641  3            !+
;  225          0642  3            ! If the number of underflows does not yet exceed the
;  226          0643  3            ! message threshold, change the severity of the condition to
;  227          0644  3            ! ERROR and resignal it so the catch-all handler will print
;  228          0645  3            ! a message and continue.  Otherwise just continue.
;  229          0646  3            !-
;  230          0647  3
;  231          0648  3            IF .UNDERFLOW_COUNT LEQ K_UNDERFLOW_THRESHOLD
;  232          0649  3            THEN
;  233          0650  4                BEGIN
;  234          0651  4                BLOCK [SIG_ARGS_ADR [CHF$L_SIG_NAME], STS$V_SEVERITY; , BYTE] = STS$K_ERROR;
;  235          0652  4                RETURN SS$_RESIGNAL
;  236          0653  4                END
;  237          0654  3            ELSE
;  238          0655  3                RETURN SS$_CONTINUE
;  239          0656  3            END
;  240          0657  3
;  241          0658  2        ELSE
;  242          0659  2
;  243          0660  2            !+
;  244          0661  2            ! Resignal the exception, since it is not an underflow.
```

```
;  245      0662  2            !-
;  246      0663  2
;  247      0664  2            RETURN SS$_RESIGNAL
;  248      0665  2
;  249      0666  1       END;                              . End of routine FOR$UNDERFLOW_HANDLER

                                                  .TITLE  FOR$UNDERF FOR$UNDERF - Fortran underflow excep
;                                                                    tion handle
                                                  .IDENT  \1-003\

                                                  .PSECT  _FOR$DATA,NOEXE,  PIC,9

                                     00000 UNDERFLOW_COUNT:
                                                  .BLKB   4
                            00000000  00004 EXIT_HNDLR_LOCK:
                                                  .LONG   0                                          ;
                                     00008 EXIT_REASON:
                                                  .BLKB   4
   0000000C  00000002  00000000  00000000  0000C EXIT_BLOCK.
                                                  .LONG   0, 0, 2, 0                                 ;

                                                  .EXTRN  LIB$SIM_TRAP, LIB$SIGNAL
                                                  .EXTRN  LIB$STOP, SYS$SETAST
                                                  .EXTRN  SYS$DCLEXH

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

                                     007C 00000          .ENTRY  FOR$UNDERFLOW_HANDLER, Save R2,R3,R4,R5,R6   ; 0548
                 56 00000000G  00   9E 00002          MOVAB   SYS$SETAST, R6                             ;
                 55 00000000'  EF   9E 00009          MOVAB   EXIT_HNDLR_LOCK, R5                        ;
                 52       04   AC   D0 00010          MOVL    SIG_ARGS_ADR, R2                           ; 0580
000004C4  8F   04  A2   1C   00   ED 00014          CMPZV   #0, #28, 4(R2), #1220                      ; 0581
                            0D   12 0001E          BNEQ    1$                                         ;
                      08   AC   DD 00020          PUSHL   MCH_ARGS_ADR                               ; 0591
                            52   DD 00023          PUSHL   R2                                         ;
            00000000G  00   02   FB 00025          CALLS   #2, LIB$SIM_TRAP                           ;
                                 04 0002C          RET                                                ;
0000049C  8F   04  A2   1C   00   ED 0002D 1$:      CMPZV   #0, #28, 4(R2), #1180                      ; 0595
                            0C   13 00037          BEQL    2$                                         ;
001682CC  8F   04  A2   1C   00   ED 00039          CMPZV   #0, #28, 4(R2), #1475276                   ; 0598
                            5D   12 00043          BNEQ    8$                                         ;
                      FC   A5   D6 00045 2$:      INCL    UNDERFLOW_COUNT                            ; 0612
                 45        65   E8 00048          BLBS    EXIT_HNDLR_LOCK, 6$                        ; 0614
                            7E   D4 0004B          CLRL    -(SP)                                      ; 0617
                 66        01   FB 0004D          CALLS   #1, SYS$SETAST                             ;
                 54        50   D0 00050          MOVL    R0, AST_STATUS                             ;
                 1D        65   E8 00053          BLBS    EXIT_HNDLR_LOCK, 3$                        ; 0618
         0C   A5   0000V   CF   9E 00056          MOVAB   EXIT_HANDLER, EXIT_BLOCK+4                 ; 0628
         14   A5      04   A5   9E 0005C          MOVAB   EXIT_REASON, EXIT_BLOCK+12                 ; 0629
                      08   A5   9F 00061          PUSHAB  EXIT_BLOCK                                 ; 0630
            00000000G  00   01   FB 00064          CALLS   #1, SYS$DCLEXH                             ;
                 53        50   D0 0006B          MOVL    R0, DCLEXH_STATUS                          ;
                 65        01   D0 0006E          MOVL    #1, EXIT_HNDLR_LOCK                        ; 0631
                            03   11 00071          BRB     4$                                         ;
                 53        01   D0 00073 3$:      MOVL    #1, DCLEXH_STATUS                          ; 0634
                 09        54   D1 00076 4$:      CMPL    AST_STATUS, #9                             ; 0636
```

```
                                                  05  12 00079          BNEQ    5$                                          ;
                                                  01  DD 0007B          PUSHL   #1                                          ;
                                            66    01  FB 0007D          CALLS   #1, SYS$SETAST                              ;
                                            0D    53  E8 00080 5$:      BLBS    DCLEXH_STATUS, 6$                           ; 0638
                                                  8F  DD 00083          PUSHL   #1540116                                    ;
                                 00000000G  00    01  FB 00089          CALLS   #1, LIB$STOP                                ;
                                            02 FC A5  D1 00090 6$:      CMPL    UNDERFLOW_COUNT, #2                         ; 0648
                                                  08  14 00094          BGTR    7$                                          ;
      04   A2            03                 00    02  F0 00096          INSV    #2, #0, #3, 4(R2)                           ; 0651
                                                  04  11 0009C          BRB     8$                                          ; 0655
                                            50    01  D0 0009E 7$:      MOVL    #1, R0                                      ;
                                                  04  000A1              RET                                                 ; 0664
                                            50 0918 8F  3C 000A2 8$:      MOVZWL  #2328, R0                                   ;
                                                  04  000A7              RET                                                 ; 0666

; Routine Size:  168 bytes,    Routine Base:  _FOR$CODE + 0000
```

```
;  251          0667  1 ROUTINE EXIT_HANDLER (
;  252          0668  1         EXIT_REASON
;  253          0669  1     ) : NOVALUE =
;  254          0670  2     BEGIN
;  255          0671  2
;  256          0672  2     !+
;  257          0673  2     ! Cause a message containing the total number of underflows to be
;  258          0674  2     ! printed if nonzero.  The severity is STS$K_INFO = 3 = INFORMATION.
;  259          0675  2     !-
;  260          0676  2
;  261          0677  2     IF .UNDERFLOW_COUNT GTR 0
;  262          0678  2     THEN
;  263          0679  2         LIB$SIGNAL (FOR$_FLOUNDEXC, 1, .UNDERFLOW_COUNT);
;  264          0680  2
;  265          0681  1     END;                                    ! End of routine EXIT_HANDLER



                                        0004 00000 EXIT_HANDLER:
                                                   .WORD   Save R2                                          ; 0667
                          52 00000000'  EF  9E 00002           MOVAB   UNDERFLOW_COUNT, R2                 ;
                                        62  D5 00009           TSTL    UNDERFLOW_COUNT                     ; 0677
                                        11  15 0000B           BLEQ    1$                                  ;
                                        62  DD 0000D           PUSHL   UNDERFLOW_COUNT                     ; 0679
                                        01  DD 0000F           PUSHL   #1                                  ;
                             00188963   8F  DD 00011           PUSHL   #1608035                            ;
                    00000000G  00       03  FB 00017           CALLS   #3, LIB$SIGNAL                      ;
                                            04 0001E 1$:       RET                                         ; 0681

; Routine Size:  31 bytes,    Routine Base:  _FOR$CODE + 00A8

;  266          0682  1
;  267          0683  1 END                                     ! End of module FOR$UNDERF
;  268          0684  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                        Bytes                       Attributes
;
;  _FOR$DATA                          28  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(9)
;  _FOR$CODE                         199  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        14         0       581          00:01.0
```

```
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:FORUNDERF/OBJ=OBJ$:FORUNDERF MSRC$:FORUNDERF/UPDATE=(ENH$:FORUNDERF
;       )

; Size:         199 code + 28 data bytes
; Run Time:       00:07.2
; Elapsed Time:   00:25.5
; Lines/CPU Min:   5668
; Lexemes/CPU-Min: 16218
; Memory Used:  77 pages
; Compilation Complete
```

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