```
M 1

BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**BASINIDFS                                            G 2





BBBBBBBB        AAAAAA        SSSSSSSS    IIIIII    NN      NN    IIIIII    DDDDDDDD    FFFFFFFFFF    SSSSSSSS
BBBBBBBB        AAAAAA        SSSSSSSS    IIIIII    NN      NN    IIIIII    DDDDDDDD    FFFFFFFFFF    SSSSSSSS
BB      BB  AA      AA  SS                  II      NN      NN      II      DD      DD  FF          SS
BB      BB  AA      AA  SS                  II      NN      NN      II      DD      DD  FF          SS
BB      BB  AA      AA  SS                  II      NNNN    NN      II      DD      DD  FF          SS
BB      BB  AA      AA  SS                  II      NNNN    NN      II      DD      DD  FF          SS
BBBBBBBB    AA      AA    SSSSSS            II      NN  NN  NN      II      DD      DD  FFFFFFFF      SSSSSS
BBBBBBBB    AA      AA    SSSSSS            II      NN  NN  NN      II      DD      DD  FFFFFFFF      SSSSSS
BB      BB  AAAAAAAAAA          SS          II      NN    NNNN      II      DD      DD  FF                  SS
BB      BB  AAAAAAAAAA          SS          II      NN    NNNN      II      DD      DD  FF                  SS
BB      BB  AA      AA          SS          II      NN      NN      II      DD      DD  FF                  SS
BB      BB  AA      AA          SS          II      NN      NN      II      DD      DD  FF                  SS      ....
BBBBBBBB    AA      AA  SSSSSSSS          IIIIII    NN      NN    IIIIII    DDDDDDDD    FF          SSSSSSSS        ....
BBBBBBBB    AA      AA  SSSSSSSS          IIIIII    NN      NN    IIIIII    DDDDDDDD    FF          SSSSSSSS        ....

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 BAS$INIT_DFS (                                 ! Initialize DEF* frame
0                 IDENT = '1-005'                       ! File: BASINIDFS.B32
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 !++
1 ! FACILITY:  BASIC-PLUS-2 Frame Support
1 !
1 ! ABSTRACT:
1 !
1 !       These routines set up and tear down frames for BASIC-PLUS-2.
1 !       Frames are used for main routines, external functions,
1 !       external subroutines, internal functions (both DEFs and DEF*s)
1 !       internal subroutines (GOSUBs) and condition handlers.
1 !
1 ! ENVIRONMENT:  VAX-11 user mode
1 !
1 ! AUTHOR: John Sauter, CREATION DATE: 10-Oct-78
1 !
1 ! MODIFIED BY:
1 !
1 !       , : VERSION
1 ! 1-001 - Original.  Just a skeleton.
1 ! 1-002 - Change LIB$S and OTS$S to STR$.  This routine is still not
1 !          implemented.  JBS 21-MAY-1979
1 ! 1-003 - Finally, code this routine, based on BAS$INIT_DEF.
1 !          JBS 03-AUG-1979
1 ! 1-004 - Remove BAS$K_WROMATPAC, not used.  JBS 19-SEP-1979
1 ! 1-005 - Fix a comment.  JBS 07-NOV-1979
1 !--
1
1 !
```

```
;   58          0058  1 !<BLF/PAGE>
```

```
!
! SWITCHES:
!

SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);

!
! LINKAGES:
!

LINKAGE
    BAS$INIT_LINK = JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2) :     !
    GLOBAL (BSF$A_MAJOR_STG = 11, BSF$A_MINOR_STG = 10, BSF$A_TEMP_STG = 9)      !
    NOPRESERVE (8, 7, 6, 5, 4, 3, 2, 1, 0);

!
! TABLE OF CONTENTS:
!

FORWARD ROUTINE
    BAS$INIT_DFS_R8 : NOVALUE BAS$INIT_LINK;    ! start DEF*

!
! INCLUDE FILES:
!

REQUIRE 'RTLIN:RTLPSECT';                       ! macros for defing psects

REQUIRE 'RTLIN:BASFRAME';                       ! Define frame structure

REQUIRE 'RTLIN:BASINARG';                       ! Define argument list

LIBRARY 'RTLSTARLE';                            ! System symbols

!
! MACROS:
!
!       NONE
!
! EQUATED SYMBOLS:
!
!       NONE
!
! PSECTS:
!
DECLARE_PSECTS (BAS);                           ! declare psects for BAS$ facility
!
! OWN STORAGE:
!
!       NONE
!
! EXTERNAL REFERENCES:
!

EXTERNAL ROUTINE
    BAS$$SIGNAL : NOVALUE,                      ! signals error
    STR$FREE1_DX,                               ! deallocates one string
```

```
;  117          0495  1      BAS$HANDLER;                                      ! handles signals
;  118          0496  1
;  119          0497  1  !+
;  120          0498  1  ! The following are the error codes used in this module.
;  121          0499  1  !-
;  122          0500  1
;  123          0501  1  EXTERNAL LITERAL
;  124          0502  1      BAS$K_TOOFEWARG : UNSIGNED (8),                   ! Too few arguments
;  125          0503  1      BAS$K_TOOMANARG : UNSIGNED (8),                   ! Too many arguments
;  126          0504  1      BAS$K_SCAFACINT : UNSIGNED (8),                   ! Scale factor interlock
;  127          0505  1      BAS$K_PROLOSSOR : UNSIGNED (8),                   ! Program lost, sorry
;  128          0506  1      BAS$K_ARGDONMAT : UNSIGNED (8),                   ! Arguments don't match
;  129          0507  1      BAS$K_NOTIMP : UNSIGNED (8);                      ! Not implemented
;  130          0508  1
```

```
1 GLOBAL ROUTINE BAS$INIT_DFS_R8 (                      ! start DEF*
1         ARGLIST,                                      ! frame parameters
1         DATA_RELOC                                    ! start of data
1     ) : NOVALUE BAS$INIT_LINK =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Set up a frame for a BASIC-PLUS-2 DEF*.  The frame is allocated
1 !       on the stack, and R10 and R9 are set up to point to it.
1 !       The argument tells how to do the allocation.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       ARGLIST.rl.v      List of information needed to set up the
1 !                         frame.  See BASIC-PLUS-2/VAX Description
1 !                         of Generated Code for details.
1 !       DATA_RELOC.ra.v Address of the major procedure's contribution
1 !                         to the data PSECT.  This is needed so that the
1 !                         argument list can be PIC.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The values of R10 and R9, which point to the automatic
1 !       storage and the temporary storage, respectively.
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Leaves lots of things on the stack for use by the compiled
1 !       BASIC-PLUS-2 code.  These things will be removed by
1 !       BAS$END_DFS_R8.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         BSF$A_MAJOR_STG : REF BLOCK [0, BYTE] FIELD (BSF$MAJOR_FRAME),
2         BSF$A_MINOR_STG : REF BLOCK [0, BYTE] FIELD (BSF$MINOR_FRAME),
2         BSF$A_TEMP_STG;
2
2     BUILTIN
2         AP,
2         FP,
2         SP;
```

```

    MAP
        ARGLIST : REF BLOCK [0, BYTE] FIELD (BAS$INIT_ARGS),    ! arg list
        AP : REF VECTOR;                                        ! caller's arg list

!+
! Define local variables as registers.  We cannot have any stack locals
! since we manipulate the stack pointer in this routine.
!-

    REGISTER
        RETURN_ADDRESS,                                         ! address to return to
        FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD),              ! pointer to FCD
        ARRAY_DESC : REF BLOCK [0, BYTE],                       ! pointer to build array descriptors
        ARRAY_INDEX;                                            ! index for array modification

!+
! Save return address because we are going to fool with the stack
!-
    RETURN_ADDRESS = ..SP;
!+
! Make sure we are passed an argument list we understand.
!-

    IF (.ARGLIST [BAS$B_IN_V_FCD] NEQ BAS$K_IN_V_FCD) THEN BAS$$SIGNAL (BAS$K_NOTIMP);

!+
! Allocate frame control data.
!-
    FMP = .FP;
    SP = .FMP - BSF$K_LENFCDDFS + %UPVAL;
!+
! LOAD Rn (R10)
!-
   BSF$A_MINOR_STG = .SP - 127;
!+
! Initialize parts of the frame control data.
!-
    FMP [BSF$A_MARK] = 0;
    FMP [BSF$A_BASE_R11] = .BSF$A_MAJOR_STG;
    FMP [BSF$A_BASE_R10] = .BSF$A_MINOR_STG;
    FMP [BSF$B_LEN_FCD] = BSF$K_LENFCDDFS;
    FMP [BSF$B_PROC_CODE] = .ARGLIST [BAS$B_IN_PROC_C];
    FMP [BSF$W_FCD_FLAGS] = .ARGLIST [BAS$W_IN_FLAGS];
    FMP [BSF$A_PROC_ID] = .ARGLIST [BAS$L_IN_PROC_I] + .DATA_RELOC;
    FMP [BSF$A_INIT_ARG] = .ARGLIST;
    FMP [BSF$L_INIT_REL] = .DATA_RELOC;
!+
! Allocate numeric scalars.  They are all initialized to zero.
!-

    INCR COUNTER FROM 1 TO .ARGLIST [BAS$L_IN_LEN_SC] DO
        BEGIN
        SP = .SP - %UPVAL;
        .SP = 0;
        END;

```

```
2 !+
2 ! Copy formals.
2 !-
2
2     DECR COUNTER FROM MIN (.ARGLIST [BAS$B_IN_NO_FML], ((.AP [0]) AND 255)) TO 1 DO
3         BEGIN
3         SP = .SP - %UPVAL;
3         .SP = .AP [.COUNTER];
2         END;
2
2 !+
2 ! Allocate and initialize descriptors.
2 !-
2     SP = .SP - .ARGLIST [BAS$L_IN_LEN_DT];
2 !+
2 ! Set ARRAY_DESC to point to the space allocated.
2 !-
2     ARRAY_DESC = .SP;
2 !+
2 ! Load the space from the template and then modify it based
2 ! on the modification table.
2 !-
2
2     INCR COUNTER FROM 0 TO ((.ARGLIST [BAS$L_IN_LEN_DT]^-2) - 1) DO
3         BEGIN
4         ARRAY_DESC [.COUNTER*%UPVAL, 0, %BPVAL, 0] = .((.ARGLIST [BAS$L_IN_DT_TMT]) +   !
3         .DATA_RELOC + (.COUNTER*%UPVAL));
2         END;
2
2 !+
2 ! Now modify the descriptors.  These are usually array descriptors.
2 !-
2
2     INCR COUNTER FROM 0 TO (.ARGLIST [BAS$L_IN_LEN_DM] - 1) DO
3         BEGIN
3         ARRAY_INDEX = .((.ARGLIST [BAS$L_IN_DT_MOD]) + .DATA_RELOC + (.COUNTER*%UPVAL));
3         BSF$A_MINOR_STG [.ARRAY_INDEX, 0, %BPVAL, 0]    !
3         = .BSF$A_MINOR_STG [.ARRAY_INDEX, 0, %BPVAL, 0] + .BSF$A_MINOR_STG;
2         END;
2
2 !+
2 ! Allocate dynamic string descriptors.
2 !-
2
2     INCR COUNTER FROM 1 TO .ARGLIST [BAS$W_IN_NO_DST] DO
3         BEGIN
3         SP = .SP - %UPVAL;
3         .SP = 0;                                 ! Pointer 0 implies not allocated.
3         SP = .SP - %UPVAL;
3         BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_D;      ! dynamic
3         BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;      ! text
3         BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0; ! length = 0
2         END;
2
2     FMP [BSF$A_STR_DESC] = .SP;
2 !+
2 ! Allocate fixed string templates.
```

```
:  303         0680  2 !-
:  304         0681  2
:  305         0682  2     INCR COUNTER FROM 1 TO .ARGLI>T [BAS$W_IN_NO_FST] DO
:  306         0683  3         BEGIN
:  307         0684  3         SP = .SP - %UPVAL;
:  308         0685  3         .SP = 0;                                ! Pointer 0 implies not allocated.
:  309         0686  3         SP = .SP - %UPVAL;
:  310         0687  3         BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_S;      ! fixed
:  311         0688  3         BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;      ! text
:  312         0689  3         BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0; ! length = 0
:  313         0690  2         END;
:  314         0691  2
:  315         0692  2 !+
:  316         0693  2 ! Allocate numeric array elements.  They are all initialized to zero.
:  317         0694  2 !-
:  318         0695  2
:  319         0696  2     INCR COUNTER FROM 1 TO (.ARGLIST [BAS$L_IN_LEN_NA]^-2) DO
:  320         0697  3         BEGIN
:  321         0698  3         SP = .SP - %UPVAL;
:  322         0699  3         .SP = 0;
:  323         0700  2         END;
:  324         0701  2
:  325         0702  2 !+
:  326         0703  2 ! Allocate temporary cells.
:  327         0704  2 !-
:  328         0705  2
:  329         0706  3     IF ((.ARGLIST [BAS$L_IN_NO_TST] NEQ 0) OR (.ARGLIST [BAS$L_IN_NO_NMT] NEQ 0))
:  330         0707  2     THEN
:  331         0708  3         BEGIN
:  332         0709  3 !+
:  333         0710  3 ! We must set up R9.  First allocate string temporaries.
:  334         0711  3 !-
:  335         0712  3
:  336         0713  3         INCR COUNTER FROM 1 TO .ARGLIST [BAS$L_IN_NO_TST] DO
:  337         0714  4             BEGIN
:  338         0715  4             SP = .SP - %UPVAL;
:  339         0716  4             .SP = 0;                            ! Pointer 0 implies not allocated.
:  340         0717  4             SP = .SP - %UPVAL;
:  341         0718  4             BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_D;  ! dynamic
:  342         0719  4             BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;  ! text
:  343         0720  4             BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0;     ! length = 0
:  344         0721  3             END;
:  345         0722  3
:  346         0723  3 !+
:  347         0724  3 ! Point R9 to the last string descriptor allocated.
:  348         0725  3 !-
:  349         0726  3         BSF$A_TEMP_STG = .SP;
:  350         0727  3 !+
:  351         0728  3 ! Now allocate numeric temporaries.
:  352         0729  3 !-
:  353         0730  3         SP = .SP - .ARGLIST [BAS$L_IN_NO_NMT];
:  354         0731  2         END;
:  355         0732  2
:  356         0733  2 !+
:  357         0734  2 ! Store R9 in the stack frame for setting up I/O lists.
:  358         0735  2 !-
:  359         0736  2     FMP [BSF$A_BASE_R9] = .BSF$A_TEMP_STG;
```

```
;  360         0737   2  !+
;  361         0738   2  ! Complete frame.
;  362         0739   2  !-
;  363         0740   2      FMP [BSF$A_BASE_SP] = .SP;
;  364         0741   2      FMP [BSF$A_HANDLER] = BAS$HANDLER;
;  365         0742   2  !+
;  366         0743   2  ! First consistency checks.
;  367         0744   2  !-
;  368         0745   2
;  369         0746   3      IF (((.AP [0]) AND 255) NEQ .ARGLIST [BAS$B_IN_NO_FML])
;  370         0747   2      THEN
;  371         0748   2  !+
;  372         0749   2  ! The number of arguments is incorrect.
;  373         0750   2  !-
;  374         0751   3          BEGIN
;  375         0752   3
;  376         0753   4          IF (((.AP [0]) AND 255) GTRU .ARGLIST [BAS$B_IN_NO_FML])
;  377         0754   3          THEN
;  378         0755   3              BAS$$SIGNAL (BAS$K_TOOMANARG)
;  379         0756   3          ELSE
;  380         0757   3              BAS$$SIGNAL (BAS$K_TOOFEWARG);
;  381         0758   3
;  382         0759   2          END;
;  383         0760   2
;  384         0761   3      IF (((.FMP [BSF$W_FCD_FLAGS]) AND (BSF$M_FCD_RSTR)) NEQ 0)
;  385         0762   2      THEN
;  386         0763   3          BEGIN
;  387         0764   3
;  388         0765   3          LOCAL
;  389         0766   3              STR_DESC_ADDR : REF BLOCK [8, BYTE];
;  390         0767   3
;  391         0768   3  !+
;  392         0769   3  ! This procedure has been marked by the compiler as returning a
;  393         0770   3  ! string result.  Be sure that there is at least one formal, and
;  394         0771   3  ! that it is a dynamic string descriptor.  If so, null its value.
;  395         0772   3  !-
;  396         0773   3
;  397         0774   3          IF (.ARGLIST [BAS$B_IN_NO_FML] LSSU 1) THEN BAS$$SIGNAL (BAS$K_TOOFEWARG);
;  398         0775   3
;  399         0776   3          STR_DESC_ADDR = AP [1];
;  400         0777   3          STR_DESC_ADDR = ..STR_DESC_ADDR;
;  401         0778   3
;  402         0779   4          IF ((.STR_DESC_ADDR [DSC$B_CLASS] NEQU DSC$K_CLASS_D) OR        !
;  403         0780   4              (.STR_DESC_ADDR [DSC$B_DTYPE] NEQU DSC$K_DTYPE_T))
;  404         0781   3          THEN
;  405         0782   3              BAS$$SIGNAL (BAS$K_ARGDONMAT);
;  406         0783   3
;  407         0784   3  !+
;  408         0785   3  ! Null the string.  This insures that, if the procedure does not reference
;  409         0786   3  ! the string, the function will have the value of the null string.
;  410         0787   3  !-
;  411         0788   3          STR$FREE1_DX (.STR_DESC_ADDR);
;  412         0789   2          END;
;  413         0790   2
;  414         0791   2  !+
;  415         0792   2  ! Put the return address back on the stack so we can return to the
;  416         0793   2  ! caller.
;  417          0794  2 !-
;  418          0795  2     SP = .SP - %UPVAL;
;  419          0796  2     .SP = .RETURN_ADDRESS;
;  420          0797  2     RETURN;
;  421          0798  1     END;                                      of BAS$INIT_DFS_R8


                                                                      .TITLE  BAS$INIT_DFS
                                                                      .IDENT  \1-005\

                                                                      .EXTRN  BAS$$SIGNAL, STR$FREE1_DX
                                                                      .EXTRN  BAS$HANDLER, BAS$K_TOOFEWARG
                                                                      .EXTRN  BAS$K_TOOMANARG
                                                                      .EXTRN  BAS$K_SCAFACINT
                                                                      .EXTRN  BAS$K_PROLOSSOR
                                                                      .EXTRN  BAS$K_ARGDONMAT
                                                                      .EXTRN  BAS$K_NOTIMP

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

                     57              51  D0 00000 BAS$INIT_DFS_R8::
                                                                      MOVL    R1, R7                                           ; 0509
                     54              50  D0 00003                     MOVL    R0, R4
                     52              6E  D0 00006                     MOVL    (SP), RETURN_ADDRESS                             ; 0585
                     01        04    A4  91 00009                     CMPB    4(ARGLIST), #1                                   ; 0590
                                     0B  13 0000D                     BEQL    1$                                               ;
                     7E       00G    8F  9A 0000F                     MOVZBL  #BAS$K_NOTIMP, -(SP)                             ;
           00000000G 00              01  FB 00013                     CALLS   #1, BAS$$SIGNAL                                  ;
                     53              5D  D0 0001A 1$:                 MOVL    FP, FMP                                          ; 0595
                     5E        D8    A3  9E 0001D                     MOVAB   -40(R3), SP                                      ; 0596
                     5A        81    AE  9E 00021                     MOVAB   -127(SP), BSF$A_MINOR_STG                        ; 0600
                               FC    A3  D4 00025                     CLRL    -4(FMP)                                          ; 0604
              F0     A3              5A  7D 00028                     MOVQ    BSF$A_MINOR_STG, -16(FMP)                        ; 0606
              E4     A3              2C  90 0002C                     MOVB    #44, -28(FMP)                                    ; 0607
              E5     A3        05    A4  90 00030                     MOVB    5(ARGLIST), -27(FMP)                             ; 0608
              E6     A3        06    A4  B0 00035                     MOVW    6(ARGLIST), -26(FMP)                             ; 0609
              E8     A3        08 B447   9E 0003A                     MOVAB   @8(ARGLIST)[DATA_RELOC], -24(FMP)                ; 0610
              D8     A3              54  D0 00040                     MOVL    ARGLIST, -40(FMP)                                ; 0611
              DC     A3              57  D0 00044                     MOVL    DATA_RELOC, -36(FMP)                             ; 0612
                                     50  D4 00048                     CLRL    COUNTER                                          ; 0617
                                     05  11 0004A                     BRB     3$                                               ;
                     5E              04  C2 0004C 2$:                 SUBL2   #4, SP                                           ; 0619
                                     6E  D4 0004F                     CLRL    (SP)                                             ; 0620
     F6              50        10    A4  F3 00051 3$:                 AOBLEQ  16(ARGLIST), COUNTER, 2$                         ; 0617
                     50        14    A4  9A 00056                     MOVZBL  20(ARGLIST), R0                                  ; 0627
                     50              6C  91 0005A                     CMPB    (AP), R0                                         ;
                                     03  1E 0005D                     BGEQU   4$                                               ;
                     50              6C  9A 0005F                     MOVZBL  (AP), R0                                         ;
                                     50  D6 00062 4$:                 INCL    COUNTER                                          ;
                                     07  11 00064                     BRB     6$                                               ;
                     5E              04  C2 00066 5$:                 SUBL2   #4, SP                                           ; 0629
                     6E            6C40  D0 00069                     MOVL    (AP)[COUNTER], (SP)                              ; 0630
                     F5              50  F5 0006D 6$:                 SOBGTR  COUNTER, 5$                                      ; 0627
                     5E        18    A4  C2 00070                     SUBL2   24(ARGLIST), SP                                  ; 0636
                     50              5E  D0 00074                     MOVL    SP, ARRAY_DESC                                   ; 0640
     56         18   A4        FE    8F  78 00077                     ASHL    #-2, 24(ARGLIST), R6                             ; 0646
                     51              01  CE 0007D                     MNEGL   #1, COUNTER                                      ; 0648
```

```
                                0A  11 00080         BRB     8$                                          ;
55           57          1C     A4  C1 00082  7$:    ADDL3   28(ARGLIST), DATA_RELOC, R5                 ; 0649
           6041               6541  D0 00087         MOVL    (R5)[COUNTER], (ARRAY_DESC)[COUNTER]        ; 0648
F2           51                 56  F2 0008C  8$:    AOBLSS  R6, COUNTER, 7$                             ; 0646
             51                 01  CE 00090         MNEGL   #1, COUNTER                                 ; 0658
                                0F  11 00093         BRB     10$                                         ;
55           57          24     A4  C1 00095  9$:    ADDL3   36(ARGLIST), DATA_RELOC, R5                 ;
             50               6541  D0 0009A         MOVL    (R5)[COUNTER], ARRAY_INDEX                  ;
                              604A  9F 0009E         PUSHAB  (ARRAY_INDEX)[BSF$A_MINOR_STG]              ; 0660
             9E                 5A  C0 000A1         ADDL2   BSF$A_MINOR_STG, @(SP)+                     ;
EC           51          20     A4  F2 000A4 10$:    AOBLSS  32(ARGLIST), COUNTER, 9$                    ; 0656
             51          28     A4  3C 000A9         MOVZWL  40(ARGLIST), R1                             ; 0667
                                50  D4 000AD         CLRL    COUNTER                                     ;
                                0F  11 000AF         BRB     12$                                         ;
             5E                 04  C2 000B1 11$:    SUBL2   #4, SP                                      ; 0669
                                6E  D4 000B4         CLRL    (SP)                                        ; 0670
             5E                 04  C2 000B6         SUBL2   #4, SP                                      ; 0671
             6E 020E0000        8F  D0 000B9         MOVL    #34471936, (SP)                             ; 0674
ED           50                 51  F3 000C0 12$:    AOBLEQ  R1, COUNTER, 11$                            ; 0667
        E0   A3                 5E  D0 000C4         MOVL    SP, -32(FMP)                                ; 0677
             51          2A     A4  3C 000C8         MOVZWL  42(ARGLIST), R1                             ; 0682
                                50  D4 000CC         CLRL    COUNTER                                     ;
                                0F  11 000CE         BRB     14$                                         ;
             5E                 04  C2 000D0 13$:    SUBL2   #4, SP                                      ; 0684
                                6E  D4 000D3         CLRL    (SP)                                        ; 0685
             5E                 04  C2 000D5         SUBL2   #4, SP                                      ; 0686
             6E 010E0000        8F  D0 000D8         MOVL    #17694720, (SP)                             ; 0689
ED           50                 51  F3 000DF 14$:    AOBLEQ  R1, COUNTER, 13$                            ; 0682
50      2C   A4          FE     8F  78 000E3         ASHL    #-2, 44(ARGLIST), R0                        ; 0696
                                51  D4 000E9         CLRL    COUNTER                                     ;
                                05  11 000EB         BRB     16$                                         ;
             5E                 04  C2 000ED 15$:    SUBL2   #4, SP                                      ; 0698
                                6E  D4 000F0         CLRL    (SP)                                        ; 0699
F7           51                 50  F3 000F2 16$:    AOBLEQ  R0, COUNTER, 15$                            ; 0696
                         30     A4  D5 000F6         TSTL    48(ARGLIST)                                 ; 0706
                                05  12 000F9         BNEQ    17$                                         ;
                         34     A4  D5 000FB         TSTL    52(ARGLIST)                                 ;
                                1F  13 000FE         BEQL    20$                                         ;
                                50  D4 00100 17$:    CLRL    COUNTER                                     ; 0713
                                0F  11 00102         BRB     19$                                         ;
             5E                 04  C2 00104 18$:    SUBL2   #4, SP                                      ; 0715
                                6E  D4 00107         CLRL    (SP)                                        ; 0716
             5E                 04  C2 00109         SUBL2   #4, SP                                      ; 0717
             6E 020E0000        8F  D0 0010C         MOVL    #34471936, (SP)                             ; 0720
EC           50          30     A4  F3 00113 19$:    AOBLEQ  48(ARGLIST), COUNTER, 18$                   ; 0713
             59                 5E  D0 00118         MOVL    SP, BSF$A_TEMP_STG                          ; 0726
             5E          34     A4  C2 0011B         SUBL2   52(ARGLIST), SP                             ; 0730
        EC   A3                 59  D0 0011F 20$:    MOVL    BSF$A_TEMP_STG, -20(FMP)                    ; 0736
        F8   A3                 5E  D0 00123         MOVL    SP, -8(FMP)                                 ; 0740
             63 00000000G       00  9E 00127         MOVAB   BAS$HANDLER, (FMP)                          ; 0741
        14   A4                 6C  91 0012E         CMPB    (AP), 20(ARGLIST)                           ; 0746
                                13  13 00132         BEQL    23$                                         ;
                                06  1B 00134         BLEQU   21$                                         ; 0753
             7E         00G     8F  9A 00136         MOVZBL  #BAS$K_TOOMANARG, -(SP)                     ; 0755
                                04  11 0013A         BRB     22$                                         ;
             7E         00G     8F  9A 0013C 21$:    MOVZBL  #BAS$K_TOOFEWARG, -(SP)                     ; 0757
   00000000G 00                 01  FB 00140 22$:    CALLS   #1, BAS$$SIGNAL                             ;
```

```
             37       E6    A3           0D  E1 00147 23$:    BBC      #13, -26(FMP), 27$                       ; 0761
                                   14    A4  95 0014C         TSTB     20(ARGLIST)                              ; 0774
                                         0B  12 0014F         BNEQ     24$                                      ;
                            7E     00G   8F  9A 00151         MOVZBL   #BAS$K_TOOFEWARG, -(SP)                  ;
                00000000G   00           01  FB 00155         CALLS    #1, BAS$$SIGNAL                          ;
                            53     04    AC  9E 0015C 24$:    MOVAB    4(AP), STR_DESC_ADDR                     ; 0776
                            53           63  D0 00160         MOVL     (STR_DESC_ADDR), STR_DESC_ADDR           ; 0777
                            02     03    A3  91 00163         CMPB     3(STR_DESC_ADDR), #2                     ; 0779
                                         06  12 00167         BNEQ     25$                                      ;
                            0E     02    A3  91 00169         CMPB     2(STR_DESC_ADDR), #14                    ; 0780
                                         0B  13 0016D         BEQL     26$                                      ;
                            7E     00G   8F  9A 0016F 25$:    MOVZBL   #BAS$K_ARGDONMAT, -(SP)                  ; 0782
                00000000G   00           01  FB 00173         CALLS    #1, BAS$$SIGNAL                          ;
                                         53  DD 0017A 26$:    PUSHL    STR_DESC_ADDR                            ; 0788
                00000000G   00           01  FB 0017C         CALLS    #1, STR$FREE1_DX                         ;
                            5E           04  C2 00183 27$:    SUBL2    #4, SP                                   ; 0795
                            6E           52  D0 00186         MOVL     RETURN_ADDRESS, (SP)                     ; 0796
                                             05 00189         RSB                                               ; 0798

; Routine Size:  394 bytes,    Routine Base:  _BAS$CODE + 0000


;  422          0799  1
;  423          0800  1 END
;  424          0801  1
;  425          0802  0 ELUDOM



;                                   PSECT SUMMARY
;
;      Name                       Bytes                       Attributes
;
;  _BAS$CODE                          394  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        6         0         581         00:01.1



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

; Size:          394 code + 0 data bytes
; Run Time:         00:12.4
; Elapsed Time:     00:24.9
; Lines/CPU Min:      3886
; Lexemes/CPU-Min: 20699
; Memory Used:  170 pages
; Compilation Complete
```

0024 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASINIGSC
LIS

BASINIT
LIS

BASINIDEF
LIS

BASINIDFS
LIS

BASINIGSB
LIS

BASINSTR
LIS

BASINIONE
LIS

BASLEFT
LIS

BASMARGIN
LIS

BASINIIOL
LIS

BASKILL
LIS

BASIOBEG
LIS

BASIOEND
LIS

BASMATADD
LIS

BASMAGTAP
LIS