```
                                                  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
```

FILEID**BASINIDEF

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

```
0 MODULE BAS$INIT_DEF (                               !
0                 IDENT = '1-012'                     ! File: BASINIDEF.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 ! 1-001 - Original.  This is just a shell.
1 ! 1-002 - Copy code from BASINIT.  JBS 07-FEB-1979
1 ! 1-003 - Remove tests for scale factors and long/double, since the
1 !          compiler will not permit a DEF to have different options
1 !          than its containing major procedure.  JBS 07-FEB-1979
1 ! 1-004 - Change from BAS$ to BSF$ for BASIC stack frame.  JBS 08-FEB-1979
1 ! 1-005 - Remove BAS$B_IN_L_FCD.  JBS 09-FEB-1979
1 ! 1-006 - When nulling the first string parameter, hold its address
1 !          in an auxiliary variable to avoid a BLISS bug.  JBS 12-FEB-1979
1 ! 1-007 - Allocate fixed string templates on the stack.  JBS 20-MAR-1979
1 ! 1-008 - Don't imply that R11 points to a frame.  JBS 08-MAY-1979
```

```
;      58          0058  1  ! 1-009 - Change LIB$S and OTS$S to STR$.  JBS 21-MAY-1979
;      59          0059  1  ! 1-010 - Use shift operator instead of divide.  JBS 11-JUN-1979
;      60          0060  1  ! 1-011 - Check for a proper argument list.  JBS 03-AUG-1979
;      61          0061  1  ! 1-012 - Remove BAS$K_WRDMATPAC, not used.  JBS 19-SEP-1979
;      62          0062  1  !--
;      63          0063  1
;      64          0064  1  .<BLF/PAGE>
```

```
:   66          0065  1 !
:   67          0066  1 ! SWITCHES:
:   68          0067  1 !
:   69          0068  1
:   70          0069  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
:   71          0070  1
:   72          0071  1 !
:   73          0072  1 ! LINKAGES:
:   74          0073  1 !
:   75          0074  1
:   76          0075  1 LINKAGE
:   77          0076  1     BAS$INIT_LINK = JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2) :     !
:   78          0077  1     GLOBAL (BSF$A_MAJOR_STG = 11, BSF$A_MINOR_STG = 10, BSF$A_TEMP_STG = 9)     !
:   79          0078  1     NOPRESERVE (8, 7, 6, 5, 4, 3, 2, 1, 0);
:   80          0079  1
:   81          0080  1 !
:   82          0081  1 ! TABLE OF CONTENTS:
:   83          0082  1 !
:   84          0083  1
:   85          0084  1 FORWARD ROUTINE
:   86          0085  1     BAS$INIT_DEF_R8 : NOVALUE BAS$INIT_LINK;    ! start DEF
:   87          0086  1
:   88          0087  1 !
:   89          0088  1 ! INCLUDE FILES:
:   90          0089  1 !
:   91          0090  1
:   92          0091  1 REQUIRE 'RTLIN:RTLPSECT';                        ! macros for defing psects
:   93          0186  1
:   94          0187  1 REQUIRE 'RTLIN:BASFRAME';                        ! Define frame structure
:   95          0390  1
:   96          0391  1 REQUIRE 'RTLIN:BASINARG';                        ! Define argument list
:   97          0475  1
:   98          0476  1 LIBRARY 'RTLSTARLE';                             ! System definitions
:   99          0477  1
:  100          0478  1 !
:  101          0479  1 ! MACROS:
:  102          0480  1 !
:  103          0481  1 !       NONE
:  104          0482  1 !
:  105          0483  1 ! EQUATED SYMBOLS:
:  106          0484  1 !
:  107          0485  1 !       NONE
:  108          0486  1 !
:  109          0487  1 ! PSECTS:
:  110          0488  1 !
:  111          0489  1 DECLARE_PSECTS (BAS);                            ! declare psects for BAS$ facility
:  112          0490  1 !
:  113          0491  1 ! OWN STORAGE:
:  114          0492  1 !
:  115          0493  1 !       NONE
:  116          0494  1 !
:  117          0495  1 ! EXTERNAL REFERENCES:
:  118          0496  1 !
:  119          0497  1
:  120          0498  1 EXTERNAL ROUTINE
:  121          0499  1     BAS$$SIGNAL : NOVALUE,                       ! signals error
:  122          0500  1     STR$FREE1_DX,                                ! Deallocates a string
```

```
;  123          0501  1      BAS$HANDLER;                                   ! handles signals
;  124          0502  1
;  125          0503  1  !
;  126          0504  1  ! The following are the error codes used in this module.
;  127          0505  1  !
;  128          0506  1
;  129          0507  1  EXTERNAL LITERAL
;  130          0508  1      BAS$K_TOOFEWARG : UNSIGNED (8),                ! Too few arguments
;  131          0509  1      BAS$K_TOOMANARG : UNSIGNED (8),                ! Too many arguments
;  132          0510  1      BAS$K_SCAFACINT : UNSIGNED (8),                ! Scale factor interlock
;  133          0511  1      BAS$K_PROLOSSOR : UNSIGNED (8),                ! Program lost, sorry
;  134          0512  1      BAS$K_ARGDONMAT : UNSIGNED (8),                ! Arguments don't match
;  135          0513  1      BAS$K_NOTIMP : UNSIGNED (8);                   ! Not implemented
;  136          0514  1
```

```
1 GLOBAL ROUTINE BAS$INIT_DEF_R8 (                          ! start DEF
1           ARGLIST,                                        ! frame parameters
1           DATA_RELOC                                      ! Start of data
1       ) : NOVALUE BAS$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.ra.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$END_DEF_R8.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         BSF$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;
```

```
2
2       MAP
2           ARGLIST : REF BLOCK [0, BYTE] FIELD (BAS$INIT_ARGS),     ! arg list
2           AP : REF VECTOR;                                  ! caller's arg list
2
2   !+
2   ! Define local variables as registers.  We cannot have any stack locals
2   ! since we manipulate the stack pointer in this routine.
2   !-
2
2       REGISTER
2           RETURN_ADDRESS,                                   ! address to return to
2           FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD),            ! pointer to FCD
2           ARRAY_DESC : REF BLOCK [0, BYTE],                 ! pointer to build array descriptors
2           ARRAY_INDEX;                                      ! index for array modification
2
2   !+
2   ! Save return address because we are going to fool with the stack
2   !-
2       RETURN_ADDRESS = ..SP;
2   !+
2   ! Make sure we are passed an argument list we understand.
2   !-
2
2       IF (.ARGLIST [BAS$B_IN_V_FCD] NEQ BAS$K_IN_V_FCD) THEN BAS$$SIGNAL (BAS$K_NOTIMP);
2
2   !+
2   ! Allocate frame control data.
2   !-
2       FMP = .FP;
2       SP = .FMP - BSF$K_LENFCDDEF + %UPVAL;
2   !+
2   ! Allocate BSF$A_USER_HAND.
2   ! It is initialized to 0 normally (ON ERROR GOTO 0), but if the
2   ! first statement in the program is ON ERROR GOTO <line number>
2   ! or ON ERROR GO BACK, it is initialized to 1 (ON ERROR GO BACK)
2   ! to prevent a "window" in which error handling is ON ERROR GOTO 0
2   ! no matter what the user wants.
2   !-
2
3       IF (((.ARGLIST [BAS$W_IN_FLAGS]) AND (BSF$M_FCD_OEGO)) NEQ 0)
2       THEN
3           BEGIN
3           SP = .SP - %UPVAL;
3           .SP = 1;
3           END
2       ELSE
3           BEGIN
3           SP = .SP - %UPVAL;
3           .SP = 0;
2           END;
2
2   !+
2   ! LOAD Rn (R10)
2   !-
2       BSF$A_MINOR_STG = .SP - 127;
2   !+
;  252          0629  2 ! Initialize parts of the frame control data.
;  253          0630  2 !-
;  254          0631  2     FMP [BSF$A_MARK] = 0;
;  255          0632  2     FMP [BSF$A_BASE_R11] = .BSF$A_MAJOR_STG;
;  256          0633  2     FMP [BSF$A_BASE_R10] = .BSF$A_MINOR_STG;
;  257          0634  2     FMP [BSF$B_LEN_FCD] = BSF$K_LENFCDDEF;
;  258          0635  2     FMP [BSF$B_PROC_CODE] = .ARGLIST [BAS$B_IN_PROC_C];
;  259          0636  2     FMP [BSF$W_FCD_FLAGS] = .ARGLIST [BAS$W_IN_FLAGS];
;  260          0637  2     FMP [BSF$A_PROC_ID] = .ARGLIST [BAS$L_IN_PROC_I] + .DATA_RELOC;
;  261          0638  2     FMP [BSF$A_INIT_ARG] = .ARGLIST;
;  262          0639  2     FMP [BSF$L_INIT_REL] = .DATA_RELOC;
;  263          0640  2 !+
;  264          0641  2 ! Allocate numeric scalars.  They are all initialized to zero.
;  265          0642  2 !-
;  266          0643  2
;  267          0644  2     INCR COUNTER FROM 1 TO .ARGLIST [BAS$L_IN_LEN_SC] DO
;  268          0645  3         BEGIN
;  269          0646  3         SP = .SP - %UPVAL;
;  270          0647  3         .SP = 0;
;  271          0648  2         END;
;  272          0649  2
;  273          0650  2 !+
;  274          0651  2 ! Copy formals.
;  275          0652  2 !-
;  276          0653  2
;  277          0654  2     DECR COUNTER FROM MIN (.ARGLIST [BAS$B_IN_NO_FML], ((.AP [0]) AND 255)) TO 1 DO
;  278          0655  3         BEGIN
;  279          0656  3         SP = .SP - %UPVAL;
;  280          0657  3         .SP = .AP [.COUNTER];
;  281          0658  2         END;
;  282          0659  2
;  283          0660  2 !+
;  284          0661  2 ! Allocate and initialize descriptors.
;  285          0662  2 !-
;  286          0663  2     SP = .SP - .ARGLIST [BAS$L_IN_LEN_DT];
;  287          0664  2 !+
;  288          0665  2 ! Set ARRAY_DESC to point to the space allocated.
;  289          0666  2 !-
;  290          0667  2     ARRAY_DESC = .SP;
;  291          0668  2 !+
;  292          0669  2 ! Load the space from the template and then modify it based
;  293          0670  2 ! on the modification table.
;  294          0671  2 !-
;  295          0672  2
;  296          0673  2     INCR COUNTER FROM 0 TO ((.ARGLIST [BAS$L_IN_LEN_DT]^-2) - 1) DO
;  297          0674  3         BEGIN
;  298          0675  4         ARRAY_DESC [.COUNTER*%UPVAL, 0, %BPVAL, 0] = .((.ARGLIST [BAS$L_IN_DT_TMT]) +   !
;  299          0676  3         .DATA_RELOC + (.COUNTER*%UPVAL));
;  300          0677  2         END;
;  301          0678  2
;  302          0679  2 !+
;  303          0680  2 ! Now modify the descriptors.  These are usually array descriptors.
;  304          0681  2 !-
;  305          0682  2
;  306          0683  2     INCR COUNTER FROM 0 TO (.ARGLIST [BAS$L_IN_LEN_DM] - 1) DO
;  307          0684  3         BEGIN
;  308          0685  3         ARRAY_INDEX = .((.ARGLIST [BAS$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.
2   !-
2
2       INCR COUNTER FROM 1 TO .ARGLIST [BAS$W_IN_NO_FST] DO
3           BEGIN
3           SP = .SP - %UPVAL;
3           .SP = 0;                                ! Pointer 0 implies not allocated.
3           SP = .SP - %UPVAL;
3           BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_S;       ! fixed
3           BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;       ! text
3           BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0; ! length = 0
2           END;
2
2   !+
2   ! Allocate numeric array elements.  They are all initialized to zero.
2   !-
2
2       INCR COUNTER FROM 1 TO (.ARGLIST [BAS$L_IN_LEN_NA]^-2) DO
3           BEGIN
3           SP = .SP - %UPVAL;
3           .SP = 0;
2           END;
2
2   !+
2   ! Allocate temporary cells.
2   !-
2
3       IF ((.ARGLIST [BAS$L_IN_NO_TST] NEQ 0) OR (.ARGLIST [BAS$L_IN_NO_NMT] NEQ 0))
2       THEN
3           BEGIN
3   !+
3   ! We must set up R9.  First allocate string temporaries.
3   !-
3
3           INCR COUNTER FROM 1 TO .ARGLIST [BAS$L_IN_NO_TST] DO
4               BEGIN
4               SP = .SP - %UPVAL;
```

```
;  4               .SP = 0;                                  ! Pointer 0 implies not allocated.
;  4               SP = .SP - %UPVAL;
;  4               BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_D;  ! dynamic
;  4               BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;  ! text
;  4               BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0;     ! length = 0
;  3               END;
;  3
;  3 !+
;  3 ! Point R9 to the last string descriptor allocated.
;  3 !-
;  3           BSF$A_TEMP_STG = .SP;
;  3 !+
;  3 ! Now allocate numeric temporaries.
;  3 !-
;  3           SP = .SP - .ARGLIST [BAS$L_IN_NO_NMT];
;  2           END;
;  2
;  2 !+
;  2 ! Store R9 in the stack frame for setting up I/O lists.
;  2 !-
;  2       FMP [BSF$A_BASE_R9] = .BSF$A_TEMP_STG;
;  2 !+
;  2 ! Complete frame.
;  2 !-
;  2       FMP [BSF$A_BASE_SP] = .SP;
;  2       FMP [BSF$A_HANDLER] = BAS$HANDLER;
;  2 !+
;  2 ! First consistency checks.
;  2 !-
;  2
;  3       IF (((.AP [0]) AND 255) NEQ .ARGLIST [BAS$B_IN_NO_FML])
;  2       THEN
;  2 !+
;  2 ! The number of arguments is incorrect.
;  2 !-
;  3           BEGIN
;  3
;  4           IF (((.AP [0]) AND 255) GTRU .ARGLIST [BAS$B_IN_NO_FML])
;  3           THEN
;  3               BAS$$SIGNAL (BAS$K_TOOMANARG)
;  3           ELSE
;  3               BAS$$SIGNAL (BAS$K_TOOFEWARG);
;  3
;  2           END;
;  2
;  3       IF (((.FMP [BSF$W_FCD_FLAGS]) AND (BSF$M_FCD_RSTR)) NEQ 0)
;  2       THEN
;  3           BEGIN
;  3
;  3           LOCAL
;  3               STR_DESC_ADDR : REF BLOCK [8, BYTE];
;  3
;  3 !+
;  3 ! This procedure has been marked by the compiler as returning a
;  3 ! string result.  Be sure that there is at least one formal, and
;  3 ! that it is a dynamic string descriptor.  If so, null its value.
;  3 !-
;  423          0800  3
;  424          0801  3          IF (.ARGLIST [BAS$B_IN_NO_FML] LSSU 1) THEN BAS$$SIGNAL (BAS$K_TOOFEWARG);
;  425          0802  3
;  426          0803  3          STR_DESC_ADDR = AP [1];
;  427          0804  3          STR_DESC_ADDR = ..STR_DESC_ADDR;
;  428          0805  3
;  429          0806  4          IF ((.STR_DESC_ADDR [DSC$B_CLASS] NEQU DSC$K_CLASS_D) OR          !
;  430          0807  4              (.STR_DESC_ADDR [DSC$B_DTYPE] NEQU DSC$K_DTYPE_T))
;  431          0808  3          THEN
;  432          0809  3              BAS$$SIGNAL (BAS$K_ARGDONMAT);
;  433          0810  3
;  434          0811  3 !+
;  435          0812  3 ! Null the string.  This insures that, if the procedure does not reference
;  436          0813  3 ! the string, the function will have the value of the null string.
;  437          0814  3 !-
;  438          0815  3          STR$FREE1_DX (.STR_DESC_ADDR);
;  439          0816  2          END;
;  440          0817  2
;  441          0818  2 !+
;  442          0819  2 ! Put the return address back on the stack so we can return to the
;  443          0820  2 ! caller.
;  444          0821  2 !-
;  445          0822  2      SP = .SP - %UPVAL;
;  446          0823  2      .SP = .RETURN_ADDRESS;
;  447          0824  2      RETURN;
;  448          0825  1      END;                                          ! of BAS$INIT_DEF_R8


                                                             .TITLE  BAS$INIT_DEF
                                                             .IDENT  \1-012\

                                                             .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$INIT_DEF_R8::
                                                             MOVL    R1, R7                                    ; 0515
                         54            50  D0 00003          MOVL    R0, R4                                    ;
                         52            6E  D0 00006          MOVL    (SP), RETURN_ADDRESS                      ; 0591
                         01      04    A4  91 00009          CMPB    4(ARGLIST), #1                            ; 0596
                                       0B  13 0000D          BEQL    1$                                        ;
                         7E      00G   8F  9A 0000F          MOVZBL  #BAS$K_NOTIMP, -(SP)                      ;
               00000000G 00            01  FB 00013          CALLS   #1, BAS$$SIGNAL                           ;
                         53            5D  D0 0001A 1$:      MOVL    FP, FMP                                   ; 0601
                         5E      D8    A3  9E 0001D          MOVAB   -40(R3), SP                               ; 0602
              08         64            3C  E1 00021          BBC     #60, (ARGLIST), 2$                        ; 0612
                         5E            04  C2 00025          SUBL2   #4, SP                                    ; 0615
                         6E            01  D0 00028          MOVL    #1, (SP)                                  ; 0616
                                       05  11 0002B          BRB     3$                                        ; 0612
                         5E            04  C2 0002D 2$:      SUBL2   #4, SP                                    ; 0620
                                       6E  D4 00030          CLRL    (SP)                                      ; 0621
```

```
                 5A        81    AE  9E 00032 3$:      MOVAB   -127(SP), BSF$A_MINOR_STG                           : 0627
                           FC    A3  D4 00036          CLRL    -4(FMP)                                             : 0631
          F0     A3              5A  7D 00039          MOVQ    BSF$A_MINOR_STG, -16(FMP)                           : 0633
          E4     A3              2C  90 0003D          MOVB    #44, -28(FMP)                                       : 0634
          E5     A3        05    A4  90 00041          MOVB    5(ARGLIST), -27(FMP)                                : 0635
          E6     A3        06    A4  B0 00046          MOVW    6(ARGLIST), -26(FMP)                                : 0636
          E8     A3        08 B447   9E 0004B          MOVAB   @8(ARGLIST)[DATA_RELOC], -24(FMP)                   : 0637
          D8     A3              54  D0 00051          MOVL    ARGLIST, -40(FMP)                                   : 0638
          DC     A3              57  D0 00055          MOVL    DATA_RELOC, -36(FMP)                                : 0639
                                 50  D4 00059          CLRL    COUNTER                                             : 0644
                                 05  11 0005B          BRB     5$                                                  :
                 5E              04  C2 0005D 4$:      SUBL2   #4, SP                                              : 0646
                                 6E  D4 00060          CLRL    (SP)                                                : 0647
     F6          50        10    A4  F3 00062 5$:      AOBLEQ  16(ARGLIST), COUNTER, 4$                            : 0644
                 50        14    A4  9A 00067          MOVZBL  20(ARGLIST), R0                                     : 0654
                 50              6C  91 0006B          CMPB    (AP), R0                                            :
                                 03  1E 0006E          BGEQU   6$                                                  :
                 50              6C  9A 00070          MOVZBL  (AP), R0                                            :
                                 50  D6 00073 6$:      INCL    COUNTER                                             :
                                 07  11 00075          BRB     8$                                                  :
                 5E              04  C2 00077 7$:      SUBL2   #4, SP                                              : 0656
                 6E            6C40  D0 0007A          MOVL    (AP)[COUNTER], (SP)                                 : 0657
                 F6              50  F5 0007E 8$:      SOBGTR  COUNTER, 7$                                         : 0654
                 5E        18    A4  C2 00081          SUBL2   24(ARGLIST), SP                                     : 0663
                 50              5E  D0 00085          MOVL    SP, ARRAY_DESC                                      : 0667
     56   18     A4        FE    8F  78 00088          ASHL    #-2, 24(ARGLIST), R6                                : 0673
                 51              01  CE 0008E          MNEGL   #1, COUNTER                                         : 0675
                                 0A  11 00091          BRB     10$                                                 :
     55          57        1C    A4  C1 00093 9$:      ADDL3   28(ARGLIST), DATA_RELOC, R5                         : 0676
               6041            6541  D0 00098          MOVL    (R5)[COUNTER], (ARRAY_DESC)[COUNTER]                : 0675
     F2          51              56  F2 0009D 10$:     AOBLSS  R6, COUNTER, 9$                                     : 0673
                 51              01  CE 000A1          MNEGL   #1, COUNTER                                         : 0685
                                 0F  11 000A4          BRB     12$                                                 :
     55          57        24    A4  C1 000A6 11$:     ADDL3   36(ARGLIST), DATA_RELOC, R5                         :
                 50            6541  D0 000AB          MOVL    (R5)[COUNTER], ARRAY_INDEX                          :
                               604A  9F 000AF          PUSHAB  (ARRAY_INDEX)[BSF$A_MINOR_STG]                      : 0687
                 9E              5A  C0 000B2          ADDL2   BSF$A_MINOR_STG, @(SP)+                             :
     EC          51        20    A4  F2 000B5 12$:     AOBLSS  32(ARGLIST), COUNTER, 11$                           : 0683
                 51        28    A4  3C 000BA          MOVZWL  40(ARGLIST), R1                                     : 0694
                                 50  D4 000BE          CLRL    COUNTER                                             :
                                 0F  11 000C0          BRB     14$                                                 :
                 5E              04  C2 000C2 13$:     SUBL2   #4, SP                                              : 0696
                                 6E  D4 000C5          CLRL    (SP)                                                : 0697
                 5E              04  C2 000C7          SUBL2   #4, SP                                              : 0698
                 6E 020E0000     8F  D0 000CA          MOVL    #34471936, (SP)                                     : 0701
     ED          50              51  F3 000D1 14$:     AOBLEQ  R1, COUNTER, 13$                                    : 0694
          E0     A3              5E  D0 000D5          MOVL    SP, -32(FMP)                                        : 0704
                 51        2A    A4  3C 000D9          MOVZWL  42(ARGLIST), R1                                     : 0709
                                 50  D4 000DD          CLRL    COUNTER                                             :
                                 0F  11 000DF          BRB     16$                                                 :
                 5E              04  C2 000E1 15$:     SUBL2   #4, SP                                              : 0711
                                 6E  D4 000E4          CLRL    (SP)                                                : 0712
                 5E              04  C2 000E6          SUBL2   #4, SP                                              : 0713
                 6E 010E0000     8F  D0 000E9          MOVL    #17694720, (SP)                                     : 0716
     ED          50              51  F3 000F0 16$:     AOBLEQ  R1, COUNTER, 15$                                    : 0709
     50   2C     A4        FE    8F  78 000F4          ASHL    #-2, 44(ARGLIST), R0                                : 0723
                                 51  D4 000FA          CLRL    COUNTER                                             :
                                       05  11 000FC          BRB      18$                                  ;
                         5E            04  C2 000FE  17$:    SUBL2    #4, SP                               ; 0725
                         6E            D4 00101              CLRL     (SP)                                 ; 0726
     F7                  51            50  F3 00103  18$:    AOBLEQ   R0, COUNTER, 17$                     ; 0723
                                   30  A4  D5 00107          TSTL     48(ARGLIST)                          ; 0733
                                       05  12 0010A          BNEQ     19$                                  ;
                                   34  A4  D5 0010C          TSTL     52(ARGLIST)                          ;
                                       1F  13 0010F          BEQL     22$                                  ;
                                       50  D4 00111  19$:    CLRL     COUNTER                              ; 0740
                                       0F  11 00113          BRB      21$                                  ;
                         5E            04  C2 00115  20$:    SUBL2    #4, SP                               ; 0742
                         6E            D4 00118              CLRL     (SP)                                 ; 0743
                         5E            04  C2 0011A          SUBL2    #4, SP                               ; 0744
                         6E  020E0000  8F  D0 0011D          MOVL     #34471936, (SP)                      ; 0747
     EC                  50        30  A4  F3 00124  21$:    AOBLEQ   48(ARGLIST), COUNTER, 20$            ; 0740
                         59            5E  D0 00129          MOVL     SP, BSF$A_TEMP_STG                   ; 0753
                         5E        34  A4  C2 0012C          SUBL2    52(ARGLIST), SP                      ; 0757
                 EC      A3            59  D0 00130  22$:    MOVL     BSF$A_TEMP_STG, -20(FMP)             ; 0763
                 F8      A3            5E  D0 00134          MOVL     SP, -8(FMP)                          ; 0767
                         63  00000000G 00  9E 00138          MOVAB    BAS$HANDLER, (FMP)                   ; 0768
                 14      A4            6C  91 0013F          CMPB     (AP), 20(ARGLIST)                    ; 0773
                                       13  13 00143          BEQL     25$                                  ;
                                       06  1B 00145          BLEQU    23$                                  ; 0780
                         7E       00G  8F  9A 00147          MOVZBL   #BAS$K_TOOMANARG, -(SP)              ; 0782
                                       04  11 0014B          BRB      24$                                  ;
                         7E       00G  8F  9A 0014D  23$:    MOVZBL   #BAS$K_TOOFEWARG, -(SP)              ; 0784
          00000000G      00            01  FB 00151  24$:    CALLS    #1, BAS$$SIGNAL                      ;
     37          E6      A3            0D  E1 00158  25$:    BBC      #13, -26(FMP), 29$                   ; 0788
                                   14  A4  95 0015D          TSTB     20(ARGLIST)                          ; 0801
                                       0B  12 00160          BNEQ     26$                                  ;
                         7E       00G  8F  9A 00162          MOVZBL   #BAS$K_TOOFEWARG, -(SP)              ;
          00000000G      00            01  FB 00166          CALLS    #1, BAS$$SIGNAL                      ;
                         53        04  AC  9E 0016D  26$:    MOVAB    4(AP), STR_DESC_ADDR                 ; 0803
                         53            63  D0 00171          MOVL     (STR_DESC_ADDR), STR_DESC_ADDR       ; 0804
                         02        03  A3  91 00174          CMPB     3(STR_DESC_ADDR), #2                 ; 0806
                                       06  12 00178          BNEQ     27$                                  ;
                         0E        02  A3  91 0017A          CMPB     2(STR_DESC_ADDR), #14                ; 0807
                                       0B  13 0017E          BEQL     28$                                  ;
                         7E       00G  8F  9A 00180  27$:    MOVZBL   #BAS$K_ARGDONMAT, -(SP)              ; 0809
          00000000G      00            01  FB 00184          CALLS    #1, BAS$$SIGNAL                      ;
                                       53  DD 0018B  28$:    PUSHL    STR_DESC_ADDR                        ; 0815
          00000000G      00            01  FB 0018D          CALLS    #1, STR$FREE1_DX                     ;
                         5E            04  C2 00194  29$:    SUBL2    #4, SP                               ; 0822
                         6E            52  D0 00197          MOVL     RETURN_ADDRESS, (SP)                 ; 0823
                                           05 0019A          RSB                                           ; 0825

; Routine Size:  411 bytes,    Routine Base:  _BAS$CODE + 0000


;   449          0826   1
;   450          0827   1 END
;   451          0828   1
;   452          0829   0 ELUDOM
;                                       PSECT SUMMARY
;
;       Name                        Bytes                         Attributes
;
;  _BAS$CODE                              411  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$:BASINIDEF/OBJ=OBJ$:BASINIDEF MSRC$:BASINIDEF/UPDATE=(ENH$:BASINIDEF
;       )

; Size:          411 code + 0 data bytes
; Run Time:         00:12.8
; Elapsed Time:     00:26.6
; Lines/CPU Min:     3901
; Lexemes/CPU-Min: 20541
; Memory Used:  176 pages
; Compilation Complete
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY