```
                                        M  1

BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      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     RRRRRRRRRRR            TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRR            TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRR            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**BASINIONE                                    I 5



BBBBBBBB      AAAAAA      SSSSSSSS    IIIIII    NN      NN    IIIIII      OOOOOO    NN      NN  EEEEEEEEEE
BBBBBBBB      AAAAAA      SSSSSSSS    IIIIII    NN      NN    IIIIII      OOOOOO    NN      NN  EEEEEEEEEE
BB      BB  AA      AA  SS              II      NN      NN      II      OO      OO  NN      NN  EE
BB      BB  AA      AA  SS              II      NN      NN      II      OO      OO  NN      NN  EE
BB      BB  AA      AA  SS              II      NNNN    NN      II      OO      OO  NNNN    NN  EE
BB      BB  AA      AA  SS              II      NNNN    NN      II      OO      OO  NNNN    NN  EE
BBBBBBBB    AA      AA    SSSSSS        II      NN  NN  NN      II      OO      OO  NN  NN  NN  EEEEEEEE
BBBBBBBB    AA      AA    SSSSSS        II      NN  NN  NN      II      OO      OO  NN  NN  NN  EEEEEEEE
BB      BB  AAAAAAAAAA          SS      II      NN    NNNN      II      OO      OO  NN    NNNN  EE
BB      BB  AAAAAAAAAA          SS      II      NN    NNNN      II      OO      OO  NN    NNNN  EE
BB      BB  AA      AA          SS      II      NN      NN      II      OO      OO  NN      NN  EE               ....
BB      BB  AA      AA          SS      II      NN      NN      II      OO      OO  NN      NN  EE               ....
BBBBBBBB    AA      AA  SSSSSSSS      IIIIII    NN      NN    IIIIII      OOOOOO    NN      NN  EEEEEEEEEE       ....
BBBBBBBB    AA      AA  SSSSSSSS      IIIIII    NN      NN    IIIIII      OOOOOO    NN      NN  EEEEEEEEEE       ....

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_ONER (
0                 IDENT = '1-003'                   ! File: BASINIONE.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.
1 ! 1-002 - Use BSF$ instead of BAS$ for stack frame prefix.  JBS 08-FEB-1979
1 ! 1-003 - Set the IV bit in the PSW if requested.  JBS 11-SEP-1979
1 !--
1
1 !<BLF/PAGE>
```

```
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1
1 LINKAGE
1     BAS$COND_JSB = JSB :                                       !
1     GLOBAL (BSF$A_MAJOR_STG = 11, BSF$A_MINOR_STG = 10, BSF$A_TEMP_STG = 9)     !
1     NOPRESERVE (8, 7, 6, 5, 4, 3, 2, 1, 0);
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     BAS$INIT_ONERR;                                    ! start condition handler
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 LIBRARY 'RTLSTARLE';                                   ! symbols for strings
1
1 REQUIRE 'RTLIN:RTLPSECT';                              ! macros for defing psects
1
1 REQUIRE 'RTLIN:BASFRAME';                              ! Define frame structure
1
1 REQUIRE 'RTLIN:BASINARG';                              ! Define frame parameters
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !       NONE
1 !
1 ! PSECTS:
1 !
1 DECLARE_PSECTS (BAS);                                  ! declare psects for BAS$ facility
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 EXTERNAL ROUTINE
1     BAS$HANDLER;                                       ! handles signals
1
```

```
1 GLOBAL ROUTINE BAS$INIT_ONERR (                         ! start condition handler
1         OLD_FMP,                                        ! frame of establisher
1         NEW_PC                                          ! where to start condition handler
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Set up a frame for a BASIC-PLUS-2 condition handler.
1 !       The frame is allocated on the stack, and R9 is left pointing
1 !       to its temporary storage.  R10 and R11 are set up from the
1 !       frame which declared the error handler.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       OLD_FMP.ra.v    Address of the frame of the establisher of
1 !                       the error handler.
1 !       NEW_PC.ra.v     Address of the first line of the condition
1 !                       handler.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The value of R9, which points to the temporary storage,
1 !       and of R10 and R11, which point to the variables of the
1 !       establisher.
1 !
1 ! ROUTINE VALUE:
1 !
1 !       The "value" of the routine is determined by how the condition
1 !       handler terminates.  See the BAS$ERROR module for the
1 !       termination routines and what value they cause to be returned.
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.  This routine calls the compiled code, and
1 !       return to this routine's caller will be made when the compiled
1 !       code does a RESUME, ON ERROR GOTO 0 or ON ERROR GO BACK.
1 !       See the BAS$ERROR module for details.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         OLD_FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD);
2
2     BUILTIN
2         FP,
2           SP,
2           BISPSW;
2
2   !+
2   ! Define local variables as registers.  We connot have any stack
2   ! locals since we manipulate the stack pointer in this routine.
2   !-
2
2       REGISTER
2           FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD),      ! pointer to FCD
2           ARGLIST : REF BLOCK [0, BYTE] FIELD (BAS$INIT_ARGS);    ! points to establisher's arg list
2
2   !+
2   ! The following registers are passed to the compiled code.
2   !-
2
2       GLOBAL REGISTER
2           BSF$A_MAJOR_STG = 11,
2           BSF$A_MINOR_STG = 10,
2           BSF$A_TEMP_STG = 9;
2
2   !+
2   ! Allocate frame control data.
2   !-
2       FMP = .FP;
2       SP = .FMP - BSF$K_LENFCDONE;
2   !+
2   ! Set up new temporary storage.
2   !-
2       ARGLIST = .OLD_FMP [BSF$A_INIT_ARG];
2
3       IF ((.ARGLIST [BAS$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;
4               BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;
4               BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0;
3               END;
3
3   !+
3   ! Point R9 to the last string descriptor allocated.
3   !-
3           BSF$A_TEMP_STG = .SP;
3   !+
3   ! Now allocate numeric temporaries.
3   !-
3           SP = .SP - .ARGLIST [BAS$L_IN_NO_NMT];
2           END;
```

```
;  226          0603  2
;  227          0604  2  !+
;  228          0605  2  ! Initialize the parts of the FCD relavent to a condition handler.
;  229          0606  2  !-
;  230          0607  2     FMP [BSF$A_MARK] = 0;
;  231          0608  2     FMP [BSF$A_BASE_SP] = .SP;
;  232          0609  2     FMP [BSF$A_BASE_R11] = (BSF$A_MAJOR_STG = .OLD_FMP [BSF$A_BASE_R11]);
;  233          0610  2     FMP [BSF$A_BASE_R10] = (BSF$A_MINOR_STG = .OLD_FMP [BSF$A_BASE_R10]);
;  234          0611  2     FMP [BSF$A_BASE_R9] = .BSF$A_TEMP_STG;
;  235          0612  2  !+
;  236          0613  2  ! The "PROCEDURE ID" is the address of the start of the condition handler.
;  237          0614  2  !-
;  238          0615  2     FMP [BSF$A_PROC_ID] = .NEW_PC;
;  239          0616  2  !+
;  240          0617  2  ! Copy the frame flags from the old frame.
;  241          0618  2  !-
;  242          0619  2     FMP [BSF$W_FCD_FLAGS] = .OLD_FMP [BSF$W_FCD_FLAGS];
;  243          0620  2  !+
;  244          0621  2  ! Set the frame ID to be "CONDITION HANDLER".  This frame ID is checked
;  245          0622  2  ! for by the RESUME, ON ERROR GOTO 0 and ON ERROR GO BACK routines.
;  246          0623  2  !-
;  247          0624  2     FMP [BSF$B_PROC_CODE] = BSF$K_PROC_ONER;
;  248          0625  2  !+
;  249          0626  2  ! Set the frame length field.
;  250          0627  2  !-
;  251          0628  2     FMP [BSF$B_LEN_FCD] = BSF$K_LENFCDONE;
;  252          0629  2  !+
;  253          0630  2  ! Set the integer interrupt enable bit in the PSW if requested.
;  254          0631  2  !-
;  255          0632  2
;  256          0633  2     IF ((.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_IV) NEQ 0) THEN BISPSW (%REF (PSW$M_IV));
;  257          0634  2
;  258          0635  2  !+
;  259          0636  2  ! Set up the exception handler.  This also marks the frame as a
;  260          0637  2  ! BASIC frame.
;  261          0638  2  !-
;  262          0639  2     FMP [BSF$A_HANDLER] = BAS$HANDLER;
;  263          0640  2  !+
;  264          0641  2  ! Branch to the compiled code.
;  265          0642  2  !-
;  266          0643  2     BAS$COND_JSB (.NEW_PC);
;  267          0644  2  !+
;  268          0645  2  ! The routine we "call" above will cut back the stack, and so never
;  269          0646  2  ! return here, but we must return a value to satisfy BLISS.
;  270          0647  2  !-
;  271          0648  2     RETURN (0);
;  272          0649  1     END;                                    ! of BAS$INIT_ONER

                                                                 .TITLE  BAS$INIT_ONER
                                                                 .IDENT  \1-003\

                                                                 .EXTRN  BAS$HANDLER

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

                                                 0FFC 00000      .ENTRY  BAS$INIT_ONERR, Save R2,R3,R4,R5,R6,R7,R8,- ; 0489
```

```
                                                                   R9,R10,R11                              :
                    50              5D  D0 00002          MOVL     FP, FMP                                 ; 0570
                    5E        E0    A0  9E 00005          MOVAB    -32(R0), SP                             ; 0571
                    52        04    AC  D0 00009          MOVL     OLD_FMP, R2                             ; 0575
                    51        D8    A2  D0 0000D          MOVL     -40(R2), ARGLIST                        :
                              30    A1  D5 00011          TSTL     48(ARGLIST)                             ; 0577
                                    05  12 00014          BNEQ     1$                                      :
                              34    A1  D5 00016          TSTL     52(ARGLIST)                             :
                                    1F  13 00019          BEQL     4$                                      :
                                    53  D4 0001B 1$:      CLRL     COUNTER                                 ; 0584
                                    0F  11 0001D          BRB      3$                                      :
                    5E              04  C2 0001F 2$:      SUBL2    #4, SP                                  ; 0586
                                    6E  D4 00022          CLRL     (SP)                                    ; 0587
                    5E              04  C2 00024          SUBL2    #4, SP                                  ; 0588
                    6E  020E0000    8F  D0 00027          MOVL     #34471936, (SP)                         ; 0591
         EC         53        30    A1  F3 0002E 3$:      AOBLEQ   48(ARGLIST), COUNTER, 2$                ; 0584
                    59              5E  D0 00033          MOVL     SP, BSF$A_TEMP_STG                      ; 0597
                    5E        34    A1  C2 00036          SUBL2    52(ARGLIST), SP                         ; 0601
                              FC    A0  D4 0003A 4$:      CLRL     -4(FMP)                                 ; 0607
              F8    A0              5E  D0 0003D          MOVL     SP, -8(FMP)                             ; 0608
                    5A        F0    A2  7D 00041          MOVQ     -16(R2), BSF$A_MINOR_STG                ; 0610
              F0    A0              5A  7D 00045          MOVQ     BSF$A_MINOR_STG, -16(FMP)               :
              EC    A0              59  D0 00049          MOVL     BSF$A_TEMP_STG, -20(FMP)                ; 0611
              E8    A0        08    AC  D0 0004D          MOVL     NEW_PC, -24(FMP)                        ; 0615
              E6    A0        E6    A2  B0 00052          MOVW     -26(R2), -26(FMP)                       ; 0619
              E4    A0      0720    8F  B0 00057          MOVW     #1824, -28(FMP)                         ; 0628
         02   E6    A0              0B  E1 0005D          BBC      #11, -26(FMP), 5$                       ; 0633
                                    20  B8 00062          BISPSW   #32                                     :
                    60  00000000G   00  9E 00064 5$:      MOVAB    BAS$HANDLER, (FMP)                      ; 0639
                              08    BC  16 0006B          JSB      @NEW_PC                                 ; 0643
                                    50  D4 0006E          CLRL     R0                                      ; 0648
                                        04 00070          RET                                              ; 0649

; Routine Size:  113 bytes,    Routine Base:  _BAS$CODE + 0000


;  273          0650  1
;  274          0651  1 END
;  275          0652  1
;  276          0653  0 ELUDOM



:                               PSECT SUMMARY
:
:      Name                     Bytes                          Attributes
:
:  _BAS$CODE                        113  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$:BASINIONE/OBJ=OBJ$:BASINIONE MSRC$:BASINIONE/UPDATE=(ENH$:BASINIONE
;       )

; Size:          113 code + 0 data bytes
; Run Time:         00:07.2
; Elapsed Time:     00:17.4
; Lines/CPU Min:     5449
; Lexemes/CPU-Min: 21137
; Memory Used:  85 pages
; Compilation Complete
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY