```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  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   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
```

```
BBBBBBBB      AAAAAA      SSSSSSSS    IIIIII    NN      NN    IIIIII      IIIIII      000000    LL
BBBBBBBB      AAAAAA      SSSSSSSS    IIIIII    NN      NN    IIIIII      IIIIII      000000    LL
BB      BB  AA      AA  SS            II      NN      NN      II          II      00      00  LL
BB      BB  AA      AA  SS            II      NN      NN      II          II      00      00  LL
BB      BB  AA      AA  SS            II      NNNN    NN      II          II      00      00  LL
BB      BB  AA      AA  SS            II      NNNN    NN      II          II      00      00  LL
BBBBBBBB    AA      AA    SSSSSS      II      NN  NN  NN      II          II      00      00  LL
BBBBBBBB    AA      AA    SSSSSS      II      NN  NN  NN      II          II      00      00  LL
BB      BB  AAAAAAAAAA          SS    II      NN    NNNN      II          II      00      00  LL
BB      BB  AAAAAAAAAA          SS    II      NN    NNNN      II          II      00      00  LL
BB      BB  AA      AA          SS    II      NN      NN      II          II      00      00  LL
BB      BB  AA      AA          SS    II      NN      NN      II          II      00      00  LL
BBBBBBBB    AA      AA  SSSSSSSS    IIIIII    NN      NN    IIIIII      IIIIII      000000    LLLLLLLLLL
BBBBBBBB    AA      AA  SSSSSSSS    IIIIII    NN      NN    IIIIII      IIIIII      000000    LLLLLLLLLL

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

```
;     1          0001  0 MODULE BAS$INIT_IOL (                          ! Initiate immediate frame
;     2          0002  0                 IDENT = '1-009'                ! File: BASINIIOL.B32 Edit: MDL1009
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************
;     7          0007  1 !*                                                                        *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                  *
;    11          0011  1 !*                                                                        *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    17          0017  1 !*  TRANSFERRED.                                                           *
;    18          0018  1 !*                                                                        *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    21          0021  1 !*  CORPORATION.                                                           *
;    22          0022  1 !*                                                                        *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !**************************************************************************
;    28          0028  1 !
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:  BASIC-PLUS-2 Frame Support
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       These routines set up and tear down frames for BASIC-PLUS-2.
;    37          0037  1 !       Frames are used for main routines, external functions,
;    38          0038  1 !       external subroutines, internal functions (both DEFs and DEF*s)
;    39          0039  1 !       internal subroutines (GOSUBs), condition handlers and immediate
;    40          0040  1 !       mode code.
;    41          0041  1 !
;    42          0042  1 ! ENVIRONMENT:  VAX-11 user mode
;    43          0043  1 !
;    44          0044  1 ! AUTHOR: John Sauter, CREATION DATE: 08-Feb-1979
;    45          0045  1 !
;    46          0046  1 ! MODIFIED BY:
;    47          0047  1 !
;    48          0048  1 ! 1-001 - Original from BAS$INIT_GOSUB.  JBS 08-Feb-1979
;    49          0049  1 ! 1-002 - Convert BAS$ to BSF$ prefixes for stack frame.  JBS 08-FEB-1979
;    50          0050  1 ! 1-003 - If the caller is not a BASIC frame, just CALL the print list.
;    51          0051  1 !         JBS 14-FEB-1979
;    52          0052  1 ! 1-004 - Change from I/O List to Immediate On-Line.  JBS 10-SEP-1979
;    53          0053  1 ! 1-005 - Set the IV bit in the PSW if reqested.  JBS 11-SEP-1979
;    54          0054  1 ! 1-006 - Give an error if the specified root frame is not a BASIC
;    55          0055  1 !         frame.  JBS 15-SEP-1979
;    56          0056  1 ! 1-007 - Correct a comment.  JBS 07-NOV-1979
;    57          0057  1 ! 1-008 - For Basic 2.0, allocate temporary strings and numerics on the
```

```
;    58          0058  1 !           stack.  PLL 29-Mar-1982
;    59          0059  1 ! 1-009 - check and make sure NO_TEMP_STR and NO_TEMP_NUM were passed as
;    60          0060  1 !           parameters before using them.  They are only passed from programs
;    61          0061  1 !           compiled with the V2.0 or later compiler.  MDL 4-Feb-1983
;    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$IOLIST_JSB = JSB :                          !
;    78          0077  1     GLOBAL (BSF$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$INIT_IOL : NOVALUE;                         ! start immediate mode code
;    87          0086  1
;    88          0087  1 !
;    89          0088  1 ! INCLUDE FILES:
;    90          0089  1 !
;    91          0090  1
;    92          0091  1 REQUIRE 'RTLIN:RTLPSECT';                           ! macros for defining psects
;    93          0186  1
;    94          0187  1 REQUIRE 'RTLIN:BASFRAME';                           ! Define frame structure
;    95          0390  1
;    96          0391  1 LIBRARY 'RTLSTARLE';                                ! Define system symbols
;    97          0392  1
;    98          0393  1 !
;    99          0394  1 ! MACROS:
;   100          0395  1 !
;   101          0396  1 !       NONE
;   102          0397  1 !
;   103          0398  1 ! EQUATED SYMBOLS:
;   104          0399  1 !
;   105          0400  1 !       NONE
;   106          0401  1 !
;   107          0402  1 ! PSECTS:
;   108          0403  1 !
;   109          0404  1 DECLARE_PSECTS (BAS);                               ! declare psects for BAS$ facility
;   110          0405  1 !
;   111          0406  1 ! OWN STORAGE:
;   112          0407  1 !
;   113          0408  1 !       NONE
;   114          0409  1 !
;   115          0410  1 ! EXTERNAL REFERENCES:
;   116          0411  1 !
;   117          0412  1
;   118          0413  1 EXTERNAL ROUTINE
;   119          0414  1     BAS$$STOP : NOVALUE,                            ! Signal a fatal BASIC error
;   120          0415  1     BAS$HANDLER;                                    ! handles signals
;   121          0416  1
;   122          0417  1 !+
```

```
;  123          0418  1 ! The following are the error codes used in this module:
;  124          0419  1 !-
;  125          0420  1
;  126          0421  1 EXTERNAL LITERAL
;  127          0422  1     BAS$K_SYNERR : UNSIGNED (8);                ! Syntax error
;  128          0423  1
```

```
;  130          0424  1 GLOBAL ROUTINE BAS$INIT_IOL (                    ! start immediate mode code
;  131          0425  1         NEW_PC,                                   ! place to start
;  132          0426  1         ROOT_FMP,                                 ! root frame
;  133          0427  1         NO_TEMP_STR,                              ! number of temporary strings
;  134          0428  1         NO_TEMP_NUM                               ! number temp. numerics (in bytes)
;  135          0429  1     ) : NOVALUE =
;  136          0430  1
;  137          0431  1 !++
;  138          0432  1 ! FUNCTIONAL DESCRIPTION:
;  139          0433  1 !
;  140          0434  1 !       Set up a frame for immediate mode code.  The frame is
;  141          0435  1 !       allocated on the stack.  R11, R10 and R9 are set up from
;  142          0436  1 !       the root's frame, which is the frame which contains the
;  143          0437  1 !       symbols known to the keyboard monitor.
;  144          0438  1 !
;  145          0439  1 ! FORMAL PARAMETERS:
;  146          0440  1 !
;  147          0441  1 !       NEW_PC.ra.v     PC of the immediate mode code.
;  148          0442  1 !       ROOT_FMP.ra.v   Address of the frame which contains the variables
;  149          0443  1 !       NO_TEMP_STR.rl.v  Number of temporary strings to allocate
;  150          0444  1 !       NO_TEMP_NUM.rl.v  Number (in bytes) of temporary numerics to allocate
;  151          0445  1 !
;  152          0446  1 ! IMPLICIT INPUTS:
;  153          0447  1 !
;  154          0448  1 !       NONE
;  155          0449  1 !
;  156          0450  1 ! IMPLICIT OUTPUTS:
;  157          0451  1 !
;  158          0452  1 !       NONE
;  159          0453  1 !
;  160          0454  1 ! ROUTINE VALUE:
;  161          0455  1 !
;  162          0456  1 !       NONE
;  163          0457  1 !
;  164          0458  1 ! COMPLETION CODES:
;  165          0459  1 !
;  166          0460  1 !       NONE
;  167          0461  1 !
;  168          0462  1 ! SIDE EFFECTS:
;  169          0463  1 !
;  170          0464  1 !       Leaves lots of things on the stack for use by the compiled
;  171          0465  1 !       BASIC-PLUS-2 code.  These things wi.  be removed by
;  172          0466  1 !       the compiled RET instruction.
;  173          0467  1 !
;  174          0468  1 !--
;  175          0469  1
;  176          0470  2     BEGIN
;  177          0471  2
;  178          0472  2     GLOBAL REGISTER
;  179          0473  2         BSF$A_MAJOR_STG = 11,
;  180          0474  2         BSF$A_MINOR_STG = 10,
;  181          0475  2         BSF$A_TEMP_STG = 9;
;  182          0476  2
;  183          0477  2     BUILTIN
;  184          0478  2         ACTUALCOUNT,
;  185          0479  2         FP,
;  186          0480  2         SP,
```

```
;  187          0481  2          BISPSW;
;  188          0482  2
;  189          0483  2      MAP
;  190          0484  2          ROOT_FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD); ! pointer to FCD
;  191          0485  2
;  192          0486  2  !+
;  193          0487  2  ! Define local variables as registers.  We connot have any stack
;  194          0488  2  ! locals since we manipulate the stack pointer in this routine.
;  195          0489  2  !-
;  196          0490  2
;  197          0491  2      REGISTER
;  198          0492  2          FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD);      ! points to our frame
;  199          0493  2
;  200          0494  2  !+
;  201          0495  2  ! If the root frame is not a BASIC frame, we have an error.
;  202          0496  2  !-
;  203          0497  2
;  204          0498  2      IF (.ROOT_FMP [BSF$A_HANDLER] NEQA BAS$HANDLER) THEN BAS$$STOP (BAS$K_SYNERR);
;  205          0499  2
;  206          0500  2  !+
;  207          0501  2  ! Load R9, R10 and R11 for the immediate mode code.
;  208          0502  2  !-
;  209          0503  2      BSF$A_MAJOR_STG = .ROOT_FMP [BSF$A_BASE_R11];
;  210          0504  2      BSF$A_MINOR_STG = .ROOT_FMP [BSF$A_BASE_R10];
;  211          0505  2  !+
;  212          0506  2  ! Allocate frame control data.
;  213          0507  2  !-
;  214          0508  2      FMP = .FP;
;  215          0509  2      SP = .FMP - BSF$K_LENFCDIOL;
;  216          0510  2  !+
;  217          0511  2  ! Allocate string temporaries on the stack.
;  218          0512  2  !-
;  219          0513  2      IF ACTUALCOUNT() GTR 2 THEN
;  220          0514  2      INCR COUNTER FROM 1 TO .NO_TEMP_STR DO
;  221          0515  3          BEGIN
;  222          0516  3          SP = .SP - %UPVAL;
;  223          0517  3          .SP = 0;                            ! ptr = 0 implies not allocated
;  224          0518  3          SP = .SP - %UPVAL;
;  225          0519  3          BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_D;
;  226          0520  3          BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;
;  227          0521  3          BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0;
;  228          0522  2          END;
;  229          0523  2  !+
;  230          0524  2  ! Point R9 to the last string descriptor allocated.
;  231          0525  2  !-
;  232          0526  2      BSF$A_TEMP_STG = .SP;
;  233          0527  2  !+
;  234          0528  2  ! Allocate temporary numerics on the stack.
;  235          0529  2  !-
;  236          0530  2      IF ACTUALCOUNT() GTR 2 THEN
;  237          0531  2      SP = .SP - .NO_TEMP_NUM;
;  238          0532  2  !+
;  239          0533  2  ! Initialize the parts of the FCD relavent to immediate mode code.
;  240          0534  2  !-
;  241          0535  2      FMP [BSF$A_MARK] = 0;
;  242          0536  2      FMP [BSF$A_BASE_SP] = .SP;
;  243          0537  2      FMP [BSF$A_BASE_R11] = .BSF$A_MAJOR_STG;
```

```
;  244          0538  2     FMP [BSF$A_BASE_R10] = .BSF$A_MINOR_STG;
;  245          0539  2     FMP [BSF$A_BASE_R9] = .BSF$A_TEMP_STG;
;  246          0540  2
;  247          0541  2 !+
;  248          0542  2 ! The "PROCEDURE ID" is the address of the start of the immediate mode code.
;  249          0543  2 !-
;  250          0544  2     FMP [BSF$A_PROC_ID] = .NEW_PC;
;  251          0545  2 !+
;  252          0546  2 ! Copy the frame flags from the root frame.
;  253          0547  2 !-
;  254          0548  2     FMP [BSF$W_FCD_FLAGS] = .ROOT_FMP [BSF$W_FCD_FLAGS];
;  255          0549  2 !+
;  256          0550  2 ! Mark this is an immediate mode frame.  The error handler will not let
;  257          0551  2 ! ON ERROR GO BACK propagate beyond this point.
;  258          0552  2 !-
;  259          0553  2     FMP [BSF$B_PROC_CODE] = BSF$K_PROC_IOL;
;  260          0554  2 !+
;  261          0555  2 ! Set the frame length field.
;  262          0556  2 !-
;  263          0557  2     FMP [BSF$B_LEN_FCD] = BSF$K_LENFCDIOL;
;  264          0558  2 !+
;  265          0559  2 ! Set the integer overflow enable bit in the PSW if the root program
;  266          0560  2 ! has integer overflow enabled.
;  267          0561  2 !-
;  268          0562  2
;  269          0563  2     IF ((.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_IV) NEQ 0) THEN BISPSW (%REF (PSW$M_IV));
;  270          0564  2
;  271          0565  2 !+
;  272          0566  2 ! Set up the handler address to mark this as a BASIC fra   and for
;  273          0567  2 ! VAX/VMS CHF.
;  274          0568  2 !-
;  275          0569  2     FMP [BSF$A_HANDLER] = BAS$HANDLER;
;  276          0570  2 !+
;  277          0571  2 ! Branch to the compiled code.  This code will issue a RET instruction
;  278          0572  2 ! rather than returning.
;  279          0573  2 !-
;  280          0574  2     BAS$IOLIST_JSB (.NEW_PC);
;  281          0575  1     END;                                    ! of BAS$INIT_IOL


                                                        .TITLE  BAS$INIT_IOL
                                                        .IDENT  \1-009\

                                                        .EXTRN  BAS$$STOP, BAS$HANDLER
                                                        .EXTRN  BAS$K_SYNERR

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

                                    0FFC 00000          .ENTRY  BAS$INIT_IOL, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0424
                                                                R10,R11
                          52      08   AC  D0 00002             MOVL    ROOT_FMP, R2                          ; 0498
                          50 00000000G 00  9E 00006             MOVAB   BAS$HANDLER, R0
                          50           62  D1 0000D             CMPL    (R2), R0
                                       0B  13 00010             BEQL    1$
                          7E      00G  8F  9A 00012             MOVZBL  #BAS$K_SYNERR, -(SP)
                00000000G 00           01  FB 00016             CALLS   #1, BAS$$STOP
                          5A      F0   A2  7D 0001D 1$:         MOVQ    -16(R2), BSF$A_MINOR_STG              ; 0504
```

```
                50              5D  D0 00021          MOVL    FP, FMP                         ; 0508
                5E        E0    AC  9E 00024          MOVAB   -32(R0), SP                     ; 0509
                02              6C  91 00028          CMPB    (AP), #2                        ; 0513
                                18  1B 0002B          BLEQU   4$
                                51  D4 0002D          CLRL    COUNTER                         ; 0514
                                0F  11 0002F          BRB     3$
                5E              04  C2 00031 2$:      SUBL2   #4, SP                          ; 0516
                                6E  D4 00034          CLRL    (SP)                            ; 0517
                5E              04  C2 00036          SUBL2   #4, SP                          ; 0518
                6E  020E0000    8F  D0 00039          MOVL    #34471936, (SP)                 ; 0521
        EC      51        0C    AC  F3 00040 3$:      AOBLEQ  NO_TEMP_STR, COUNTER, 2$        ; 0514
                59              5E  D0 00045 4$:      MOVL    SP, BSF$A_TEMP_STG              ; 0526
                02              6C  91 00048          CMPB    (AP), #2                        ; 0530
                                04  1B 0004B          BLEQU   5$
                5E        10    AC  C2 0004D          SUBL2   NO_TEMP_NUM, SP                 ; 0531
                          FC    A0  D4 00051 5$:      CLRL    -4(FMP)                         ; 0535
          F8    A0              5E  D0 00054          MOVL    SP, -8(FMP)                     ; 0536
          F0    A0              5A  7D 00058          MOVQ    BSF$A_MINOR_STG, -16(FMP)       ; 0538
          EC    A0              59  D0 0005C          MOVL    BSF$A_TEMP_STG, -20(FMP)        ; 0539
          E8    A0        04    AC  D0 00060          MOVL    NEW_PC, -24(FMP)                ; 0544
          E6    A0        E6    A2  B0 00065          MOVW    -26(R2), -26(FMP)               ; 0548
          E4    A0      0820    8F  B0 0006A          MOVW    #2080, -28(FMP)                 ; 0557
        02  E6    A0              0B  E1 00070          BBC     #11, -26(FMP), 6$               ; 0563
                                20  B8 00075          BISPSW  #32
                60 00000000G    00  9E 00077 6$:      MOVAB   BAS$HANDLER, (FMP)              ; 0569
                          04    BC  16 0007E          JSB     @NEW_PC                         ; 0574
                                    04 00081          RET                                     ; 0575

; Routine Size:  130 bytes,    Routine Base:  _BAS$CODE + 0000

;  282          0576  1
;  283          0577  1 END
;  284          0578  1
;  285          0579  0 ELUDOM


;                                  PSECT SUMMARY

;      Name                    Bytes                           Attributes

; _BAS$CODE                      130  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$:BASINIIOL/OBJ=OBJ$:BASINIIOL MSRC$:BASINIIOL/UPDATE=(ENH$:BASINIIOL
;       )

; Size:         130 code + 0 data bytes
; Run Time:        00:06.7
; Elapsed Time:    00:13.3
; Lines/CPU Min:    5185
; Lexemes/CPU-Min: 19244
; Memory Used:  83 pages
; Compilation Complete
```

BASINIGSC
LIS

BASINIT
LIS

BASINIDEF
LIS

BASINIDFS
LIS

BASINIGSB
LIS

BASINSTR
LIS

BASINIONE
LIS

BASLEFT
LIS

BASMARGIN
LIS

BASINIIOL
LIS

BASKILL
LIS

BASTOBEG
LIS

BASTOEND
LIS

BASMATADD
LIS

BASMAGTAP
LIS