```
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
```

FILEID**BASSTOP

```
BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS  TTTTTTTTTT    000000    PPPPPPPP
BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS  TTTTTTTTTT    000000    PPPPPPPP
BB      BB  AA      AA  SS          SS              TT      00      00  PP      PP
BB      BB  AA      AA  SS          SS              TT      00      00  PP      PP
BB      BB  AA      AA  SS          SS              TT      00      00  PP      PP
BB      BB  AA      AA  SS          SS              TT      00      00  PP      PP
BBBBBBBB    AA      AA    SSSSSS      SSSSSS        TT      00      00  PPPPPPPP
BBBBBBBB    AA      AA    SSSSSS      SSSSSS        TT      00      00  PPPPPPPP
BB      BB  AAAAAAAAAA          SS          SS      TT      00      00  PP
BB      BB  AAAAAAAAAA          SS          SS      TT      00      00  PP
BB      BB  AA      AA          SS          SS      TT      00      00  PP              ....
BB      BB  AA      AA          SS          SS      TT      00      00  PP              ....
BBBBBBBB    AA      AA  SSSSSSSS    SSSSSSSS        TT        000000    PP              ....
BBBBBBBB    AA      AA  SSSSSSSS    SSSSSSSS        TT        000000    PP              ....

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$STOP (                                    !
;     2          0002  0                 IDENT = '1-006'                     ! File: BASSTOP.B32 EDIT: PL1006
;     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 ! FACILITY:  BASIC-PLUS-2 Miscellaneous
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains the BASIC STOP statement, which prompts
;    36          0036  1 !       the user to EXIT or CONTINUE.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  VAX-11 User Mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: 10-MAY-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.
;    45          0045  1 ! 1-002 - Change LIB$S and OTS$S to STR$.  JBS 21-MAY-1979
;    46          0046  1 ! 1-003 - Use LIB$GET_COMMAND instead of doing BASIC I/O.  JBS 14-SEP-1979
;    47          0047  1 ! 1-004 - Add BAS$$STOP_INIT, for the RUN command.  JBS 15-SEP-1979
;    48          0048  1 ! 1-005 - If just a return is typed then just signal WHAT?. FM 5-FEB-81.
;    49          0049  1 ! 1-006 - LIB$STOP should be declared EXTERNAL.  PLL 20-Nov-81
;    50          0050  1 !--
;    51          0051  1
;    52          0052  1 !<BLF/PAGE>
```

```
;    54          0053  1 !
;    55          0054  1 ! SWITCHES:
;    56          0055  1 !
;    57          0056  1
;    58          0057  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    59          0058  1
;    60          0059  1 !
;    61          0060  1 ! LINKAGES:
;    62          0061  1 !
;    63          0062  1 !       NONE
;    64          0063  1 !
;    65          0064  1 ! TABLE OF CONTENTS:
;    66          0065  1 !
;    67          0066  1
;    68          0067  1 FORWARD ROUTINE
;    69          0068  1     BAS$STOP : NOVALUE,                     ! EXIT or CONTINUE
;    70          0069  1     BAS$$STOP_INIT : NOVALUE,               ! Set up for the RUN environment
;    71          0070  1     FREE_STRINGS;                           ! Handler to free strings
;    72          0071  1
;    73          0072  1 !
;    74          0073  1 ! INCLUDE FILES:
;    75          0074  1 !
;    76          0075  1
;    77          0076  1 REQUIRE 'RTLIN:RTLPSECT';                   ! Macros for defining psects
;    78          0171  1
;    79          0172  1 LIBRARY 'RTLSTARLE';                        ! System symbols
;    80          0173  1
;    81          0174  1 !
;    82          0175  1 ! MACROS:
;    83          0176  1 !
;    84          0177  1 !       NONE
;    85          0178  1 !
;    86          0179  1 ! EQUATED SYMBOLS:
;    87          0180  1 !
;    88          0181  1 !       NONE
;    89          0182  1 !
;    90          0183  1 ! PSECTS:
;    91          0184  1 !
;    92          0185  1 DECLARE_PSECTS (BAS);                       ! Declare psects for BAS$ facility
;    93          0186  1 !
;    94          0187  1 ! OWN STORAGE:
;    95          0188  1 !
;    96          0189  1
;    97          0190  1 OWN
;    98          0191  1     RUN_CMD : INITIAL (0);                  ! 1 means we are in the RUN environment.
;    99          0192  1
;   100          0193  1 !
;   101          0194  1 ! EXTERNAL REFERENCES:
;   102          0195  1 !
;   103          0196  1
;   104          0197  1 EXTERNAL ROUTINE
;   105          0198  1     LIB$STOP : NOVALUE,                     ! Signal fatal error
;   106          0199  1     LIB$GET_COMMAND,                        ! Read a command
;   107          0200  1     BAS$$SIGNAL : NOVALUE,                  ! Signal an error
;   108          0201  1     STR$FREE1_DX,                           ! Free a string
;   109          0202  1     LIB$MATCH_COND,                         ! Match condition values
;   110          0203  1     BAS$$STOP : NOVALUE;                    ! Signal a fatal error
```

```
;  111          0204  1
;  112          0205  1 !+
;  113          0206  1 ! The following are the error codes used in this module.
;  114          0207  1 !-
;  115          0208  1
;  116          0209  1 EXTERNAL LITERAL
;  117          0210  1     BAS$K_STO : UNSIGNED (8),                   ! Stop
;  118          0211  1     BAS$K_WHA : UNSIGNED (8);                   ! What?
;  119          0212  1
;  120          0213  1 !+
;  121          0214  1 ! The following code is used in the call to $EXIT
;  122          0215  1 !-
;  123          0216  1
;  124          0217  1 EXTERNAL LITERAL
;  125          0218  1     BAS$_STO;                                   ! Stop
;  126          0219  1
```

```
;  128          0220  1 GLOBAL ROUTINE BAS$STOP : NOVALUE =                ! Exit or Continue
;  129          0221  1
;  130          0222  1 !++
;  131          0223  1 ! FUNCTIONAL DESCRIPTION:
;  132          0224  1 !
;  133          0225  1 !       Prompt the user to EXIT or CONTINUE his program.  On entry  a
;  134          0226  1 !       message is signalled to show the line number of the STOP
;  135          0227  1 !       statement.  If the user elects to exit, we call SYS$EXIT.
;  136          0228  1 !
;  137          0229  1 ! FORMAL PARAMETERS:
;  138          0230  1 !
;  139          0231  1 !       NONE
;  140          0232  1 !
;  141          0233  1 ! IMPLICIT INPUTS:
;  142          0234  1 !
;  143          0235  1 !       NONE
;  144          0236  1 !
;  145          0237  1 ! IMPLICIT OUTPUTS:
;  146          0238  1 !
;  147          0239  1 !       NONE
;  148          0240  1 !
;  149          0241  1 ! ROUTINE VALUE:
;  150          0242  1 ! COMPLETION CODES:
;  151          0243  1 !
;  152          0244  1 !       NONE
;  153          0245  1 !
;  154          0246  1 ! SIDE EFFECTS:
;  155          0247  1 !
;  156          0248  1 !       May never return to its caller.
;  157          0249  1 !
;  158          0250  1 !--
;  159          0251  1
;  160          0252  2     BEGIN
;  161          0253  2
;  162          0254  2     LOCAL
;  163          0255  2         PROMPT_DESC : BLOCK [8, BYTE],          ! The prompt
;  164          0256  2         A_DESC : BLOCK [8, BYTE] VOLATILE,      ! String to receive message
;  165          0257  2         A_BUF : REF VECTOR [65535, BYTE],       ! Characters of that string
;  166          0258  2         EXIT_OR_CONT;                           ! Flag for what we are to do
;  167          0259  2
;  168          0260  2 !+
;  169          0261  2 ! Enable a handler to free our string in case we are running under
;  170          0262  2 ! the RUN command, and the run-time environment elects not to
;  171          0263  2 ! return here after our STOP signal.
;  172          0264  2 !-
;  173          0265  2
;  174          0266  2     ENABLE
;  175          0267  2         FREE_STRINGS (A_DESC);
;  176          0268  2
;  177          0269  2 !+
;  178          0270  2 ! Set up the descriptors.
;  179          0271  2 !-
;  180          0272  2     PROMPT_DESC [DSC$W_LENGTH] = %CHARCOUNT ('#');
;  181          0273  2     PROMPT_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  182          0274  2     PROMPT_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  183          0275  2     PROMPT_DESC [DSC$A_POINTER] = UPLIT (%ASCII'#');
;  184          0276  2     A_DESC [DSC$W_LENGTH] = 0;
```

```
;  185          0277  2      A_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  186          0278  2      A_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
;  187          0279  2      A_DESC [DSC$A_POINTER] = 0;
;  188          0280  2  !+
;  189          0281  2  ! Print a message giving the line number of the STOP statement.
;  190          0282  2  ! This is done using the signalling facility.
;  191          0283  2  !-
;  192          0284  2      BAS$$SIGNAL (BAS$K_STO);
;  193          0285  2  !+
;  194          0286  2  ! Now we wait for a typein to indicate whether to exit or continue.
;  195          0287  2  ! This is suppressed in the RUN environment.
;  196          0288  2  !-
;  197          0289  2
;  198          0290  3      IF ( NOT .RUN_CMD)
;  199          0291  2      THEN
;  200          0292  3          BEGIN
;  201          0293  3
;  202          0294  3          DO
;  203          0295  4              BEGIN
;  204          0296  4
;  205          0297  4              LOCAL
;  206          0298  4                  OUT_LEN,
;  207          0299  4                  GET_COM_STATU
;  208          0300  4
;  209          0301  4  !+
;  210          0302  4  ! Ask whether or exit or continue.  The prompt is a #.
;  211          0303  4  !-
;  212          0304  4              GET_COM_STATUS = LIB$GET_COMMAND (A_DESC, PROMPT_DESC, OUT_LEN);
;  213          0305  4
;  214          0306  4              IF ( NOT .GET_COM_STATUS) THEN LIB$STOP (.GET_COM_STATUS);
;  215          0307  4
;  216          0308  4  !+
;  217          0309  4  ! If just a return is typed then sigal WHAT.
;  218          0310  4  ! If the first character of the response is a C, we are to continue.
;  219          0311  4  ! If an E, we are to exit.  Otherwise, print an error message and
;  220          0312  4  ! ask again.
;  221          0313  4  !-
;  222          0314  4
;  223          0315  4              IF .OUT_LEN EQL 0
;  224          0316  4              THEN
;  225          0317  5                  BEGIN
;  226          0318  5                  BAS$$SIGNAL (BAS$K_WHA);
;  227          0319  5                  EXIT_OR_CONT = 0;
;  228          0320  5                  END
;  229          0321  4              ELSE
;  230          0322  5                  BEGIN
;  231          0323  5                  A_BUF = .A_DESC [DSC$A_POINTER];
;  232          0324  5
;  233          0325  5                  SELECTONE .A_BUF [0] OF
;  234          0326  5                      SET
;  235          0327  5
;  236          0328  5                      [%C'C', %C'c'] :
;  237          0329  5                          EXIT_OR_CONT = 1;
;  238          0330  5
;  239          0331  5                      [%C'E', %C'e'] :
;  240          0332  5                          EXIT_OR_CONT = 2;
;  241          0333  5
```

```
;  242     0334  5                        [OTHERWISE] :
;  243     0335  6                            BEGIN
;  244     0336  6 !+
;  245     0337  6 ! Print the WHAT message by signaling.
;  246     0338  6 !-
;  247     0339  6                            BAS$$SIGNAL (BAS$K_WHA);
;  248     0340  6                            EXIT_OR_CONT = 0;
;  249     0341  5                            END;
;  250     0342  5                        TES;
;  251     0343  5
;  252     0344  4                    END;
;  253     0345  4
;  254     0346  4                END
;  255     0347  3            UNTIL (.EXIT_OR_CONT NEQ 0);
;  256     0348  3
;  257     0349  3 !+
;  258     0350  3 ! When we get here, we have received either an EXIT or CONTINUE
;  259     0351  3 ! command.  First free our string.
;  260     0352  3 !-
;  261     0353  3            STR$FREE1_DX (A_DESC);
;  262     0354  3 !+
;  263     0355  3 ! If the user said to exit, do so.
;  264     0356  3 !-
;  265     0357  3
;  266     0358  3            IF (.EXIT_OR_CONT EQL 2) THEN $EXIT (CODE = BAS$_STO);
;  267     0359  3
;  268     0360  2            END;
;  269     0361  2
;  270     0362  2 !+
;  271     0363  2 ! Otherwise, return to our caller.
;  272     0364  2 !-
;  273     0365  2     RETURN;
;  274     0366  1     END;                                    ! end of BAS$STOP


                                                            .TITLE  BAS$STOP
                                                            .IDENT  \1-006\

                                                            .PSECT  _BAS$DATA,NOEXE,  PIC,2

                                00000000  00000 RUN_CMD:    .LONG   0                                       ;

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

                            00  00  00  23  00000 P.AAA:    .ASCII  \#\<0><0><0>                            ;

                                                            .EXTRN  LIB$STOP, LIB$GET_COMMAND
                                                            .EXTRN  BAS$$SIGNAL, STR$FREE1_DX
                                                            .EXTRN  LIB$MATCH_COND, BAS$$STOP
                                                            .EXTRN  BAS$K_STO, BAS$K_WHA
                                                            .EXTRN  BAS$_STO, SYS$EXIT

                                   001C 00000               .ENTRY  BAS$STOP, Save R2,R3,R4                 ; 0220
                    54 00000000G   00   9E 00002            MOVAB   BAS$$SIGNAL, R4                         ;
                    5E             14   C2 00009            SUBL2   #20, SP                                 ;
                           04      AE   7C 0000C            CLRQ    A_DESC                                  ; 0252
                    6D       0097  CF   DE 0000F            MOVAL   95, (FP)                                ;
```

```
                0C   AE 010E0001    8F   D0 00014          MOVL    #17694721, PROMPT_DESC                  ; 0272
                10   AE       DD    AF   9E 0001C          MOVAB   P.AAA, PROMPT_DESC+4                    ; 0275
                              04    AE   B4 00021          CLRW    A_DESC                                  ; 0276
                06   AE              0E   90 00024          MOVB    #14, A_DESC+2                           ; 0277
                07   AE              02   90 00028          MOVB    #2, A_DESC+3                            ; 0278
                              08    AE   D4 0002C          CLRL    A_DESC+4                                ; 0279
                     7E       00G   8F   9A 0002F          MOVZBL  #BAS$K_STO, -(SP)                       ; 0284
                     64              01   FB 00033          CALLS   #1, BAS$$SIGNAL
                     6C 00000000'    EF   E8 00036          BLBS    RUN_CMD, 8$                             ; 0290
                                     5E   DD 0003D 1$:      PUSHL   SP                                      ; 0304
                              10    AE   9F 0003F          PUSHAB  PROMPT_DESC
                              0C    AE   9F 00042          PUSHAB  A_DESC
          00000000G           00        03   FB 00045          CALLS   #3, LIB$GET_COMMAND
                              09        50   E8 0004C          BLBS    GET_COM_STATUS, 2$                  ; 0306
                                        50   DD 0004F          PUSHL   GET_COM_STATUS
          00000000G           00        01   FB 00051          CALLS   #1, LIB$STOP
                                        6E   D5 00058 2$:      TSTL    OUT_LEN                             ; 0315
                                        26   13 0005A          BEQL    6$
                              53        08   AE   D0 0005C          MOVL    A_DESC+4, A_BUF                 ; 0323
                        43    8F              63   91 00060          CMPB    (A_BUF), #67                    ; 0328
                                              06   13 00064          BEQL    3$
                        63    8F              63   91 00066          CMPB    (A_BUF), #99
                                              05   12 0006A          BNEQ    4$
                              52              01   D0 0006C 3$:      MOVL    #1, EXIT_OR_CONT                ; 0329
                                              1A   11 0006F          BRB     7$
                        45    8F              63   91 00071 4$:      CMPB    (A_BUF), #69                    ; 0331
                                              06   13 00075          BEQL    5$
                        65    8F              63   91 00077          CMPB    (A_BUF), #101
                                              05   12 0007B          BNEQ    6$
                              52              02   D0 0007D 5$:      MOVL    #2, EXIT_OR_CONT                ; 0332
                                              09   11 00080          BRB     7$
                              7E       00G    8F   9A 00082 6$:      MOVZBL  #BAS$K_WHA, -(SP)               ; 0339
                              64              01   FB 00086          CALLS   #1, BAS$$SIGNAL
                                              52   D4 00089          CLRL    EXIT_OR_CONT                    ; 0340
                                              B0   13 0008B 7$:      BEQL    1$                              ; 0347
                                        04    AE   9F 0008D          PUSHAB  A_DESC                          ; 0353
          00000000G           00        01   FB 00090          CALLS   #1, STR$FREE1_DX
                              02        52   D1 00097          CMPL    EXIT_OR_CONT, #2                    ; 0358
                                        0D   12 0009A          BNEQ    8$
                                  00000000G   8F   DD 0009C          PUSHL   #BAS$_STO
          00000000G           00        01   FB 000A2          CALLS   #1, SYS$EXIT
                                        04   000A9 8$:         RET                                         ; 0366
                                     0000   000AA 9$:         .WORD   Save nothing                        ; 0252
                              50        08   AC   D0 000AC          MOVL    8(AP), R0
                              50        04   A0   D0 000B0          MOVL    4(R0), R0
                                        F0   A0   9F 000B4          PUSHAB  A_DESC
                                        01   DD 000B7          PUSHL   #1
                                        5E   DD 000B9          PUSHL   SP
                              7E        04   AC   7D 000BB          MOVQ    4(AP), -(SP)
                        0000V CF              03   FB 000BF          CALLS   #3, FREE_STRINGS
                                              04   000C4          RET

; Routine Size:  197 bytes,    Routine Base:  _BAS$CODE + 0004

;  275          0367  1
```

```
;  277          0368  1 GLOBAL ROUTINE BAS$$STOP_INIT : NOVALUE =       ! Set up for RUN command
;  278          0369  1
;  279          0370  1 !++
;  280          0371  1 ! FUNCTIONAL DESCRIPTION:
;  281          0372  1 !
;  282          0373  1 !       Set up for the RUN environment.  Since this image is to run under the RUN
;  283          0374  1 !       command, the keyboard monitor will intercept the STOP call (by trapping
;  284          0375  1 !       the signal) and will return only if and when the program is to continue.
;  285          0376  1 !       Therefore, do not ask (redundently) for Exit or Continue.
;  286          0377  1 !
;  287          0378  1 ! FORMAL PARAMETERS:
;  288          0379  1 !
;  289          0380  1 !       NONE
;  290          0381  1 !
;  291          0382  1 ! IMPLICIT INPUTS:
;  292          0383  1 !
;  293          0384  1 !       NONE
;  294          0385  1 !
;  295          0386  1 ! IMPLICIT OUTPUTS:
;  296          0387  1 !
;  297          0388  1 !       RUN_CMD.wb
;  298          0389  1 !
;  299          0390  1 ! ROUTINE VALUE:
;  300          0391  1 ! COMPLETION CODES:
;  301          0392  1 !
;  302          0393  1 !       NONE
;  303          0394  1 !
;  304          0395  1 ! SIDE EFFECTS:
;  305          0396  1 !
;  306          0397  1 !       Disables the normal dialog.
;  307          0398  1 !
;  308          0399  1 !--
;  309          0400  1
;  310          0401  2     BEGIN
;  311          0402  2 !+
;  312          0403  2 ! Flag that we are in the RUN environment.  This will prevent the
;  313          0404  2 ! normal dialog on STOP.
;  314          0405  2 !-
;  315          0406  2     RUN_CMD = 1;
;  316          0407  2     RETURN;
;  317          0408  1     END;                                    ! end of BAS$$RUN_INIT


                                              0000 00000          .ENTRY  BAS$$STOP_INIT, Save nothing                ; 0368
                        00000000'  EF       01   D0 00002         MOVL    #1, RUN_CMD                                 ; 0406
                                              04 00009          RET                                                 ; 0408

; Routine Size:  10 bytes,    Routine Base:  _BAS$CODE + 00C9

;  318          0409  1
```

```
;  320     0410  1 ROUTINE FREE_STRINGS (                          ! Free local strings
;  321     0411  1         SIG,                                    ! Signal vector
;  322     0412  1         MECH,                                   ! Mechanism vector
;  323     0413  1         ENBL                                    ! Enable vector
;  324     0414  1     ) =
;  325     0415  1
;  326     0416  1 !++
;  327     0417  1 ! FUNCTIONAL DESCRIPTION:
;  328     0418  1 !
;  329     0419  1 !       If we are unwinding, free the local strings.  They are passed
;  330     0420  1 !       in the enable vector.
;  331     0421  1 !
;  332     0422  1 ! FORMAL PARAMETERS:
;  333     0423  1 !
;  334     0424  1 !       SIG.rl.a        A counted vector of parameters to LIB$SIGNAL/STOP
;  335     0425  1 !       MECH.rl.a       A counted vector of info from CHF
;  336     0426  1 !       ENBL.ra.a       A counted vector of ENABLE argument addresses.
;  337     0427  1 !
;  338     0428  1 ! IMPLICIT INPUTS:
;  339     0429  1 !
;  340     0430  1 !       NONE
;  341     0431  1 !
;  342     0432  1 ! IMPLICIT OUTPUTS:
;  343     0433  1 !
;  344     0434  1 !       NONE
;  345     0435  1 !
;  346     0436  1 ! ROUTINE VALUE:
;  347     0437  1 ! COMPLETION CODES:
;  348     0438  1 !
;  349     0439  1 !       Always SS$_RESIGNAL, which is ignored when unwinding.
;  350     0440  1 !
;  351     0441  1 ! SIDE EFFECTS:
;  352     0442  1 !
;  353     0443  1 !       Frees all of the strings passed as enable arguments.
;  354     0444  1 !
;  355     0445  1 !--
;  356     0446  1
;  357     0447  2     BEGIN
;  358     0448  2
;  359     0449  2     MAP
;  360     0450  2         SIG : REF VECTOR,
;  361     0451  2         MECH : REF VECTOR,
;  362     0452  2         ENBL : REF VECTOR;
;  363     0453  2
;  364     0454  2 !+
;  365     0455  2 ! Only free the strings if this is the UNWIND condition.
;  366     0456  2 !-
;  367     0457  2
;  368     0458  2     IF ( NOT (LIB$MATCH_COND (SIG [1], %REF (SS$_UNWIND)))) THEN RETURN (SS$_RESIGNAL);
;  369     0459  2
;  370     0460  2 !+
;  371     0461  2 ! Go through the enable arguments, freeing them.
;  372     0462  2 !-
;  373     0463  2
;  374     0464  2     INCR ARG_NO FROM 1 TO .ENBL [0] DO
;  375     0465  3         BEGIN
;  376     0466  3         STR$FREE1_DX (.ENBL [.ARG_NO]);
```

```
;  377          0467  2          END;
;  378          0468  2
;  379          0469  2      RETURN (SS$_RESIGNAL);
;  380          0470  1      END;                                    ! end of FREE_STRINGS



                                        0004 00000 FREE_STRINGS:
                                                   .WORD   Save R2                                  ; 0410
                            7E      0920   8F  3C 00002           MOVZWL  #2336, -(SP)             ; 0458
                                           5E  DD 00007           PUSHL   SP                       ;
                      7E        04   AC       04  C1 00009        ADDL3   #4, SIG, -(SP)           ;
                 00000000G      00           02  FB 0000E        CALLS   #2, LIB$MATCH_COND       ;
                                14           50  E9 00015        BLBC    R0, 3$                   ;
                                             52  D4 00018        CLRL    ARG_NO                   ; 0464
                                             0B  11 0001A        BRB     2$                       ;
                                          0C BC42  DD 0001C 1$:  PUSHL   @ENBL[ARG_NO]            ; 0466
                 00000000G      00           01  FB 00020        CALLS   #1, STR$FREE1_DX         ;
                      F0                52   0C  BC  F3 00027 2$:  AOBLEQ  @ENBL, ARG_NO, 1$       ; 0464
                                        50 0918  8F  3C 0002C 3$:  MOVZWL  #2328, R0                ; 0469
                                                 04 00031          RET                            ; 0470

; Routine Size:  50 bytes,    Routine Base:  _BAS$CODE + 00D3

;  381          0471  1 END                                        ! end of module BAS$STOP
;  382          0472  1
;  383          0473  0 ELUDOM
;
;                                  PSECT SUMMARY
;
;      Name                    Bytes                           Attributes
;
;  _BAS$DATA                      4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _BAS$CODE                    261  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         11         0          581        00:01.1
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         257 code + 8 data bytes
; Run Time:       00:07.0
; Elapsed Time:   00:14.7
; Lines/CPU Min:   4054
; Lexemes/CPU-Min: 13277
; Memory Used:  74 pages
; Compilation Complete
```

BASUDFWF
LIS

BASSTRING
LIS

BASTERMIO
LIS

BASTRM
LIS

BASTAB
LIS

BASUDFRM
LIS

BASSYS
LIS

BASSTR
LIS

BASUDFRL
LIS