```
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
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**BASUPDATE

```
BBBBBBBB      AAAAAA      SSSSSSSS  UU      UU  PPPPPPPP    DDDDDDDD      AAAAAA    TTTTTTTTTT  EEEEEEEEEE
BBBBBBBB      AAAAAA      SSSSSSSS  UU      UU  PPPPPPPP    DDDDDDDD      AAAAAA    TTTTTTTTTT  EEEEEEEEEE
BB      BB   AA      AA  SS        UU      UU  PP      PP  DD      DD   AA      AA      TT      EE
BB      BB   AA      AA  SS        UU      UU  PP      PP  DD      DD   AA      AA      TT      EE
BB      BB   AA      AA  SS        UU      UU  PP      PP  DD      DD   AA      AA      TT      EE
BB      BB   AA      AA  SS        UU      UU  PP      PP  DD      DD   AA      AA      TT      EE
BBBBBBBB     AA      AA   SSSSSS   UU      UU  PPPPPPPP    DD      DD   AA      AA      TT      EEEEEEEE
BBBBBBBB     AA      AA   SSSSSS   UU      UU  PPPPPPPP    DD      DD   AA      AA      TT      EEEEEEEE
BB      BB   AAAAAAAAAA         SS  UU      UU  PP          DD      DD   AAAAAAAAAA      TT      EE
BB      BB   AAAAAAAAAA         SS  UU      UU  PP          DD      DD   AAAAAAAAAA      TT      EE
BB      BB   AA      AA         SS  UU      UU  PP          DD      DD   AA      AA      TT      EE                ....
BB      BB   AA      AA         SS  UU      UU  PP          DD      DD   AA      AA      TT      EE                ....
BBBBBBBB     AA      AA  SSSSSSSS   UUUUUUUUUU  PP          DDDDDDDD     AA      AA      TT      EEEEEEEEEE        ....
BBBBBBBB     AA      AA  SSSSSSSS   UUUUUUUUUU  PP          DDDDDDDD     AA      AA      TT      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
```

```
;     1          0001  0 MODULE BAS$UPDATE (      ! Basic UPDATE construct
;     2          0002  0                    IDENT = '1-007'          ! File: BASUPDATE.B32 Edit:MDL1007
;     3          0003  0                    ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !*****************************************************************************
;     8          0008  1 !*                                                                           *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                     *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18          0018  1 !*  TRANSFERRED.                                                             *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22          0022  1 !*  CORPORATION.                                                             *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:
;    33          0033  1 !       Basic support library - user callable
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !       This module is the UPI level of the Basic UPDATE construct.  Initially,
;    37          0037  1 !       it contains only the code for sequential I/O.  This module will set
;    38          0038  1 !       up the I/O data base for the LUN and go directly to the REC level.
;    39          0039  1 !
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:
;    42          0042  1 !       User access mode - AST reentrant.
;    43          0043  1 !
;    44          0044  1 ! AUTHOR: Donald G. Petersen, CREATION DATE: 27-Feb-79
;    45          0045  1 !
;    46          0046  1 ! MODIFIED BY:
;    47          0047  1 !
;    48          0048  1 !       DGP, 27-Feb-79 : VERSION 01
;    49          0049  1 ! 1-001 - original.  DGP 27-Feb-79
;    50          0050  1 ! 1-002 - Add Update with count.  DGP 28-Feb-79
;    51          0051  1 ! 1-003 - Add Addressing switches.  DGP 18-Jun-79
;    52          0052  1 ! 1-004 - Check for virtual array use and read only.  DGP 16-Oct-79
;    53          0053  1 ! 1-005 - If no count is specified, use the current record size, not the
;    54          0054  1 !         buffer size.  DGP 12-Feb-80
;    55          0055  1 ! 1-006 - If channel is not already open signal. FM 12-FEB-81.
;    56          0056  1 ! 1-007 - Set up ISB$A_USER_FP.  MDL 8-Nov-1983
;    57          0057  1 !--
```

```
;    59          0058  1 !
;    60          0059  1 ! SWITCHES
;    61          0060  1 !
;    62          0061  1
;    63          0062  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    64          0063  1
;    65          0064  1 !
;    66          0065  1 ! LINKAGES
;    67          0066  1 !
;    68          0067  1         REQUIRE 'RTLIN:OTSLNK';         ! Define all linkages
;    69          0496  1
;    70          0497  1 !
;    71          0498  1 ! TABLE OF CONTENTS:
;    72          0499  1 !
;    73          0500  1
;    74          0501  1 FORWARD ROUTINE
;    75          0502  1         BAS$UPDATE_COUN:NOVALUE,                ! UPI UPDATE with count
;    76          0503  1         BAS$UPDATE:NOVALUE;             ! UPI level Sequential UPDATE
;    77          0504  1
;    78          0505  1 !
;    79          0506  1 ! INCLUDE FILES:
;    80          0507  1 !
;    81          0508  1         REQUIRE 'RTLML:OTSISB';         ! ISB definitions
;    82          0676  1         REQUIRE 'RTLML:OTSLUB';         ! LUB definitions
;    83          0816  1         REQUIRE 'RTLIN:RTLPSECT';       ! Define DECLARE_PSECTS macro
;    84          0911  1         LIBRARY 'RTLSTARLE';            ! Starlet system macros
;    85          0912  1
;    86          0913  1 !
;    87          0914  1 ! MACROS:
;    88          0915  1 !
;    89          0916  1 !       NONE
;    90          0917  1
;    91          0918  1 !
;    92          0919  1 ! EQUATED SYMBOLS:
;    93          0920  1 !       NONE
;    94          0921  1 !
;    95          0922  1 !
;    96          0923  1 ! PSECT DECLARATIONS:
;    97          0924  1 !
;    98          0925  1         DECLARE_PSECTS(BAS);
;    99          0926  1 !
;   100          0927  1 ! OWN STORAGE:
;   101          0928  1 !
;   102          0929  1 !       NONE
;   103          0930  1
;   104          0931  1 !
;   105          0932  1 ! EXTERNAL REFERENCES:
;   106          0933  1 !
;   107          0934  1
;   108          0935  1 EXTERNAL LITERAL
;   109          0936  1     BAS$K_IO_CHANOT : UNSIGNED (8),             ! I/O channel not open
;   110          0937  1     BAS$K_ILLILLACC : UNSIGNED (8);             ! illegal or illogical access
;   111          0938  1
;   112          0939  1 EXTERNAL ROUTINE
;   113          0940  1         BAS$$STOP_IO : NOVALUE,                 ! Signal I/O errors
;   114          0941  1         BAS$$REC_OPD:JSB_DO_WRITE NOVALUE,      ! REC level processing - RMS interface
;   115          0942  1                                         ! UPDATE sequential
```

```
;  116          0943  1        BAS$$CB_PUSH:JSB_CB_PUSH NOVALUE,
;  117          0944  1                                 ! Create LUB/ISB/RAB and push down I/O system
;  118          0945  1        BAS$$CB_POP:JSB_CB_POP NOVALUE; ! Pop I/O system back one CB
;  119          0946  1
```

```
;  121     0947  1 GLOBAL ROUTINE BAS$UPDATE (      ! UPDATE sequential
;  122     0948  1         UNIT                     ! logical unit number
;  123     0949  1         ) :NOVALUE =
;  124     0950  1
;  125     0951  1 !++
;  126     0952  1 ! FUNCTIONAL DESCRIPTION:
;  127     0953  1 !
;  128     0954  1 !       This routine will set up the I/O data base for this LUN if necessary
;  129     0955  1 !       and then go directly to the REC level.  When control is returned to
;  130     0956  1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
;  131     0957  1 !       face to RMS is done at the REC level.  The current record is updated.
;  132     0958  1 !
;  133     0959  1 ! FORMAL PARAMETERS:
;  134     0960  1 !
;  135     0961  1 !       UNIT.rlu.v      logical unit number
;  136     0962  1 !
;  137     0963  1 ! IMPLICIT INPUTS:
;  138     0964  1 !
;  139     0965  1 !       LUB$V_VA_USE            indicates virtual array use of the file
;  140     0966  1 !       LUB$V_READ_ONLY         file is read only
;  141     0967  1 !       RAB$W_RSZ               the current buffer size
;  142     0968  1 !
;  143     0969  1 ! IMPLICIT OUTPUTS:
;  144     0970  1 !
;  145     0971  1 !       ISB$B_STTM_TYPE         the statement type
;  146     0972  1 !       LUB$V_BLK_USE           indicates non-virtual array use of file
;  147     0973  1 !
;  148     0974  1 ! COMPLETION CODES:
;  149     0975  1 !
;  150     0976  1 !       NONE
;  151     0977  1 !
;  152     0978  1 ! SIDE EFFECTS:
;  153     0979  1 !
;  154     0980  1 !       Signals:
;  155     0981  1 !       BAS$K_ILLILLACC         illegal of illogical access
;  156     0982  1 !
;  157     0983  1 !--
;  158     0984  1
;  159     0985  2     BEGIN
;  160     0986  2     GLOBAL REGISTER
;  161     0987  2         CCB = K_CCB_REG: REF BLOCK[, BYTE];
;  162     0988  2
;  163     0989  2     BUILTIN
;  164     0990  2         FP;
;  165     0991  2
;  166     0992  2     LOCAL
;  167     0993  2         FMP : REF BLOCK [, BYTE];
;  168     0994  2
;  169     0995  2     FMP = .FP;
;  170     0996  2
;  171     0997  2     !+
;  172     0998  2     ! Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
;  173     0999  2     ! trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
;  174     1000  2     !-
;  175     1001  2
;  176     1002  2     BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
;  177     1003  2
```

```
;  178          1004  2     CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  179          1005  2
;  180          1006  2     !+
;  181          1007  2     ! Now that the data base is in place, store the statement type and  go
;  182          1008  2     ! directly to the REC level.
;  183          1009  2     !-
;  184          1010  2
;  185          1011  2     CCB[ISB$B_STTM_TYPE] = ISB$K_ST_TY_UPD;
;  186          1012  2 !+
;  187          1013  2 ! Check for virtual array usage and set block usage
;  188          1014  2 ! If channel not already open, signal.
;  189          1015  2 !-
;  190          1016  2     IF .CCB [LUB$V_VA_USE] OR .CCB [LUB$V_READ_ONLY] THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  191          1017  2     IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  192          1018  2
;  193          1019  2
;  194          1020  2     CCB [LUB$V_BLK_USE] = 1;
;  195          1021  2     BAS$$REC_UPD(.CCB[RAB$W_RSZ]);
;  196          1022  2
;  197          1023  2     !+
;  198          1024  2     ! Now that the UPDATE has been done, pop the CCB off the I/O system.
;  199          1025  2     !-
;  200          1026  2
;  201          1027  2     BAS$$CB_POP ();
;  202          1028  2
;  203          1029  2
;  204          1030  2
;  205          1031  1     END;                            !End of BAS$UPDATE

                                                            .TITLE  BAS$UPDATE
                                                            .IDENT  \1-007\

                                                            .EXTRN  BAS$K_IO_CHANOT
                                                            .EXTRN  BAS$K_ILLILLACC
                                                            .EXTRN  BAS$$STOP_IO, BAS$$REC_UPD
                                                            .EXTRN  BAS$$CB_PUSH, BAS$$CB_POP

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

                                    083C 00000              .ENTRY  BAS$UPDATE, Save R2,R3,R4,R5,R11        ; 0947
                    54 00000000G    00   9E 00002          MOVAB   BAS$$STOP_IO, R4                        ;
                    53              5D   D0 00009          MOVL    FP, FMP                                 ; 0995
                    50              08   CE 0000C          MNEGL   #8, R0                                  ; 1002
                    52        04    AC   D0 0000F          MOVL    UNIT, R2                                ;
                    00000000G       00   16 00013          JSB     BAS$$CB_PUSH                            ;
            FF4C    CB        0C    A3   D0 00019          MOVL    12(FMP), -180(CCB)                      ; 1004
            FF71    CB              23   90 0001F          MOVB    #35, -143(CCB)                          ; 1011
                    05        FF    AB   E8 00024          BLBS    -1(CCB), 1$                             ; 1016
07          FC    AB                02   E1 00028          BBC     #2, -4(CCB), 2$                         ;
                    7E        00G   8F   9A 0002D 1$:      MOVZBL  #BAS$K_ILLILLACC, -(SP)                 ;
                    64              01   FB 00031          CALLS   #1, BAS$$STOP_IO                        ;
                    07        FC    AB   E8 00034 2$:      BLBS    -4(CCB), 3$                             ; 1017
                    7E        00G   8F   9A 00038          MOVZBL  #BAS$K_IO_CHANOT, -(SP)                 ;
                    64              01   FB 0003C          CALLS   #1, BAS$$STOP_IO                        ;
                FF  AB                02   88 0003F 3$:      BISB2   #2, -1(CCB)                             ; 1020
                    50        22    AB   3C 00043          MOVZWL  34(CCB), R0                             ; 1021
```

```
                00000000G  00  16 00047         JSB     BAS$$REC_UPD                            ;
                00000000G  00  16 0004D         JSB     BAS$$CB_POP                             ; 1027
                           04 00053             RET                                             ; 1031

; Routine Size:  84 bytes,    Routine Base:  _BAS$CODE + 0000
```

```
;  207          1032  1 GLOBAL ROUTINE BAS$UPDATE_COUN (          ! UPDATE with count
;  208          1033  1         UNIT,                    ! logical unit number
;  209          1034  1         COUNT                    ! No. of bytes in record
;  210          1035  1         ) :NOVALUE =
;  211          1036  1
;  212          1037  1 !++
;  213          1038  1 ! FUNCTIONAL DESCRIPTION:
;  214          1039  1 !
;  215          1040  1 !       This routine will set up the I/O data base for this LUN if necessary
;  216          1041  1 !       and then go directly to the REC level.  When control is returned to
;  217          1042  1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
;  218          1043  1 !       face to RMS is done at the REC level.  The current record is updated.
;  219          1044  1 !
;  220          1045  1 ! FORMAL PARAMETERS:
;  221          1046  1 !
;  222          1047  1 !       UNIT.rlu.v      logical unit number
;  223          1048  1 !       COUNT.rl.v      No. of bytes in the record
;  224          1049  1 !
;  225          1050  1 ! IMPLICIT INPUTS:
;  226          1051  1 !
;  227          1052  1 !       LUB$V_VA_USE            virtual array use of file
;  228          1053  1 !       LUB$V_READ_ONLY         file is read only
;  229          1054  1 !
;  230          1055  1 ! IMPLICIT OUTPUTS:
;  231          1056  1 !
;  232          1057  1 !       ISB$B_STTM_TYPE         the statement type
;  233          1058  1 !       LUB$V_BLK_USE           non-virtual array use of file
;  234          1059  1 !
;  235          1060  1 ! COMPLETION CODES:
;  236          1061  1 !
;  237          1062  1 !       NONE
;  238          1063  1 !
;  239          1064  1 ! SIDE EFFECTS:
;  240          1065  1 !
;  241          1066  1 !       Signals:
;  242          1067  1 !       BAS$K_ILLILLACC         illegal or illogical access
;  243          1068  1 !
;  244          1069  1 !--
;  245          1070  1
;  246          1071  2     BEGIN
;  247          1072  2     GLOBAL REGISTER
;  248          1073  2         CCB = K_CCB_REG: REF BLOCK[, BYTE];
;  249          1074  2
;  250          1075  2     BUILTIN
;  251          1076  2         FP;
;  252          1077  2
;  253          1078  2     LOCAL
;  254          1079  2         FMP : REF BLOCK [, BYTE];
;  255          1080  2
;  256          1081  2     FMP = .FP;
;  257          1082  2
;  258          1083  2     !+
;  259          1084  2     ! Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
;  260          1085  2     ! trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
;  261          1086  2     !-
;  262          1087  2
;  263          1088  2     BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
```

```
;  264          1089  2
;  265          1090  2      CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  266          1091  2
;  267          1092  2      !+
;  268          1093  2      ! Now that the data base is in place, store the statement type and  go
;  269          1094  2      ! directly to the REC level.
;  270          1095  2      !-
;  271          1096  2
;  272          1097  2      CCB[ISB$B_STTM_TYPE] = ISB$K_ST_TY_UPD;
;  273          1098  2  !+
;  274          1099  2  ! Check for virtual array usage and set block usage
;  275          1100  2  !-
;  276          1101  2      IF .CCB [LUB$V_VA_USE] OR .CCB [LUB$V_READ_ONLY] THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  277          1102  2      CCB [LUB$V_BLK_USE] = 1;
;  278          1103  2      BAS$$REC_UPD(.COUNT);
;  279          1104  2
;  280          1105  2      !+
;  281          1106  2      ! Now that the UPDATE has been done, pop the CCB off the I/O system.
;  282          1107  2      !-
;  283          1108  2
;  284          1109  2      BAS$$CB_POP ();
;  285          1110  2
;  286          1111  2
;  287          1112  2
;  288          1113  1      END;                            !End of BAS$UPDATE


                                        083C 00000          .ENTRY  BAS$UPDATE_COUN, Save R2,R3,R4,R5,R11       ; 1032
                                  53    5D   D0 00002          MOVL    FP, FMP                                     ; 1081
                                  50    08   CE 00005          MNEGL   #8, R0                                      ; 1088
                                  52 04   AC   D0 00008          MOVL    UNIT, R2                                    ;
                          00000000G  00   16 0000C              JSB     BAS$$CB_PUSH                                ;
                       FF4C       CB 0C   A3   D0 00012          MOVL    12(FMP), -180(CCB)                          ; 1090
                       FF71       CB      23   90 00018          MOVB    #35, -143(CCB)                              ; 1097
                                  05 FF   AB   E8 0001D          BLBS    -1(CCB), 1$                                 ; 1101
                    0B       FC   AB      02   E1 00021          BBC     #2, -4(CCB), 2$                             ;
                                  7E 00G  8F   9A 00026  1$:     MOVZBL  #BAS$K_ILLILLACC, -(SP)                     ;
                    00000000G     00      01   FB 0002A          CALLS   #1, BAS$$STOP_IO                            ;
                          FF      AB      02   88 00031  2$:     BISB2   #2, -1(CCB)                                 ; 1102
                                  50 08   AC   D0 00035          MOVL    COUNT, R0                                   ; 1103
                          00000000G  00   16 00039              JSB     BAS$$REC_UPD                                ;
                          00000000G  00   16 0003F              JSB     BAS$$CB_POP                                 ; 1109
                                          04 00045              RET                                                 ; 1113

; Routine Size:  70 bytes,    Routine Base:  _BAS$CODE + 0054
```

```
;  290          1114  1 END                                    !End of module - BAS$UPDATE
;  291          1115  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                        Bytes                         Attributes
;
;  _BAS$CODE                             154  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         2         0          581         00:01.2
```

```
;                                    COMMAND QUALIFIERS

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

; Size:          154 code + 0 data bytes
; Run Time:         00:09.7
; Elapsed Time:     00:19.4
; Lines/CPU Min:      6896
; Lexemes/CPU-Min: 40843
; Memory Used:  115 pages
; Compilation Complete
```

BASUDFWL
LIS

BASUNLOCK
LIS

BASVECTOR
LIS

BASVIRTUA
LIS

BASVAL
LIS

BASVIRTIO
LIS

BASUNWIND
LIS

BASUPDATE
LIS

BASVECTR2
LIS