```
BBBBBBBB    PPPPPPPP      AAAAAA    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  BBBBBBBB   LL             KK       KK
BBBBBBBB    PPPPPPPP      AAAAAA    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  BBBBBBBB   LL             KK       KK
BB      BB  PP      PP  AA      AA  GG        EE              TT      BB      BB LL             KK     KK
BB      BB  PP      PP  AA      AA  GG        EE              TT      BB      BB LL             KK    KK
BB      BB  PP      PP  AA      AA  GG        EE              TT      BB      BB LL             KK  KK
BB      BB  PP      PP  AA      AA  GG        EE              TT      BB      BB LL             KK KK
BBBBBBBB    PPPPPPPP    AA      AA  GG        EEEEEEEE        TT      BBBBBBBB   LL             KKKKK
BBBBBBBB    PPPPPPPP    AA      AA  GG        EEEEEEEE        TT      BBBBBBBB   LL             KKKKK
BB      BB  PP         AAAAAAAAAA  GG  GGGGGG EE              TT      BB      BB LL             KK KK
BB      BB  PP         AAAAAAAAAA  GG  GGGGGG EE              TT      BB      BB LL             KK  KK
BB      BB  PP         AA      AA  GG      GG EE              TT      BB      BB LL             KK    KK
BB      BB  PP         AA      AA  GG      GG EE              TT      BB      BB LL             KK     KK      ....
BBBBBBBB    PP         AA      AA    GGGGGG   EEEEEEEEEE      TT      BBBBBBBB   LLLLLLLLLL     KK       KK    ....
BBBBBBBB    PP         AA      AA    GGGGGG   EEEEEEEEEE      TT      BBBBBBBB   LLLLLLLLLL     KK       KK    ....


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 !
     2    0002  0 !<blf/width:80>
     3    0003  0 !
     4    0004  0 MODULE BPA$GETFREBLK (                    ! Dynamic allocation of storage
     5    0005  0            IDENT = '1-311'                 ! File: BPAGETBLK.B32
     6    0006  0            ) =
     7    0007  1 BEGIN
     8    0008  1 !
     9    0009  1 !********************************************************************
    10    0010  1 !*                                                                  *
    11    0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                         *
    12    0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.          *
    13    0013  1 !*  ALL RIGHTS RESERVED.                                            *
    14    0014  1 !*                                                                  *
    15    0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
    16    0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
    17    0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
    18    0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
    19    0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
    20    0020  1 !*  TRANSFERRED.                                                     *
    21    0021  1 !*                                                                  *
    22    0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
    23    0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
    24    0024  1 !*  CORPORATION.                                                     *
    25    0025  1 !*                                                                  *
    26    0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
    27    0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.          *
    28    0028  1 !*                                                                  *
    29    0029  1 !*                                                                  *
    30    0030  1 !********************************************************************
    31    0031  1 !
    32    0032  1 !<blf/uppercase_key>
    33    0033  1 !<blf/lowercase_user>
    34    0034  1 !
    35    0035  1 !
    36    0036  1 !++
    37    0037  1 ! FACILITY: BASIC-PLUS AME
    38    0038  1 !
    39    0039  1 ! ABSTRACT:
    40    0040  1 !
    41    0041  1 !     This module contains the code to allocate and deallocate
    42    0042  1 !     blocks of heap storage.
    43    0043  1 !
    44    0044  1 ! ENVIRONMENT: Native mode VAX processor, User mode, Common RTL.
    45    0045  1 !
    46    0046  1 ! AUTHOR: Jeremy Barker, CREATION DATE: 16-Jan-79
    47    0047  1 !
    48    0048  1 ! MODIFIED BY:
    49    0049  1 !
    50    0050  1 !     VERSION X01
    51    0051  1 !
    52    0052  1 !     Viveka Eriksson, 17-Aug-79
    53    0053  1 ! 309  - Modifications to comply with VAX RTL standards.
    54    0054  1 !
    55    0055  1 !     Jeremy Barker, 17-Aug-79
    56    0056  1 ! 310  - Require REQ:ame so that generated code is in proper PSECT
    57    0057  1 !        and is sharable (also makes image 1 page smaller)
```

BPA$GETFREBLK
1-311

M 13
16-Sep-1984 01:37:58   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:56:51   [BASRTL.SRC]BPAGETBLK.B32;1

Page   2
        (1)

```
:    58        0058  1 ! 1-311 - Replace reference to require file with DISPATCH_PSECTS macro.
:    59        0059  1 !           JBS 27-SEP-1979
:    60        0060  1 !
:    61        0061  1 !--
:    62        0062  1
:    63        0063  1 !
:    64        0064  1 !<blf/page>
```

BPA$GETFREBLK
1-311

N 13
16-Sep-1984 01:37:58   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:56:51   [BASRTL.SRC]BPAGETBLK.B32;1

Page 3
      (2)

```
  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  !  TABLE OF CONTENTS:
  74      0073   1  !
  75      0074   1
  76      0075   1  FORWARD ROUTINE
  77      0076   1      bpa$get_block,                              ! allocate a block
  78      0077   1      bpa$free_block;                             ! deallocate a block
  79      0078   1
  80      0079   1  !
  81      0080   1  !  INCLUDE FILES:
  82      0081   1  !
  83      0082   1
  84      0083   1  REQUIRE 'RTLIN:RTLPSECT';                       ! Define DECLARE_PSECTS macro
  85      0178   1
  86      0179   1  !
  87      0180   1  !  MACROS:
  88      0181   1  !
  89      0182   1  !      NONE
  90      0183   1  !
  91      0184   1  !  EQUATED SYMBOLS:
  92      0185   1  !
  93      0186   1  !      NONE
  94      0187   1  !
  95      0188   1  !  PSECTS:
  96      0189   1  !
  97      0190   1  declare_psects (bpa);                           ! Declare psects for BPA$ facility
  98      0191   1  !  OWN STORAGE:
  99      0192   1  !
 100      0193   1  !      NONE
 101      0194   1  !
 102      0195   1  !
 103      0196   1  !  EXTERNAL REFERENCES:
 104      0197   1  !
 105      0198   1
 106      0199   1  EXTERNAL ROUTINE
 107      0200   1      lib$get_vm : ADDRESSING_MODE (GENERAL),
 108      0201   1                                      ! allocate a heap storage block
 109      0202   1      lib$free_vm : ADDRESSING_MODE (GENERAL);
 110      0203   1
 111      0204   1                                      ! deallocate heap storage block
 112      0205   1  !
 113      0206   1
```

```
 115    0207  1  GLOBAL ROUTINE bpa$get_block (block_size, BLOCK) =      ! allocate a block
 116    0208  1                                          ! M 309
 117    0209  1
 118    0210  1  !++
 119    0211  1  ! FUNCTIONAL DESCRIPTION:
 120    0212  1  !
 121    0213  1  !     A block of heap storage, 'block_size' bytes long, is allocated.
 122    0214  1  !     It is zeroed and its address is returned.
 123    0215  1  !
 124    0216  1  ! FORMAL PARAMETERS:
 125    0217  1  !
 126    0218  1  !     block_size - The length of the block to allocate, in bytes.
 127    0219  1  !     block      - Pointer to a longword to receive the starting
 128    0220  1  !                  address of the block allocated.
 129    0221  1  !
 130    0222  1  ! IMPLICIT INPUTS:
 131    0223  1  !
 132    0224  1  !     NONE
 133    0225  1  !
 134    0226  1  ! IMPLICIT OUTPUTS:
 135    0227  1  !
 136    0228  1  !     NONE
 137    0229  1  !
 138    0230  1  ! ROUTINE VALUE:
 139    0231  1  !
 140    0232  1  !     success - 1
 141    0233  1  !     failure - ??
 142    0234  1  !
 143    0235  1  ! SIDE EFFECTS:
 144    0236  1  !
 145    0237  1  !     A block of heap storage is allocated.
 146    0238  1  !     The size of the virtual image P0 region may be increased.
 147    0239  1  !
 148    0240  1  !--
 149    0241  1
 150    0242  2      BEGIN
 151    0243  2
 152    0244  2      MAP                                          ! A 309
 153    0245  2          BLOCK : REF VECTOR [1, LONG];    ! address of block allocated    ! A 309
 154    0246  2
 155    0247  2      LOCAL
 156    0248  2          sts;                                     ! status value of lib$get_vm
 157    0249  2
 158    0250  3      IF (sts = lib$get_vm (block_size, BLOCK [0]))        ! M 309
 159    0251  2      THEN
 160    0252  2          CH$FILL (0, .block_size, CH$PTR (.BLOCK [0]));
 161    0253  2
 162    0254  2                                          ! Zero the allocated block ! M 309
 163    0255  2      RETURN .sts;                                 ! M 309
 164    0256  1      END;                                 !End of bpa$get_block
```

                                        .TITLE   BPA$GETFREBLK
                                        .IDENT   \1-311\

                                        .EXTRN   LIBGET_VM, LIBFREE_VM

BPA$GETFREBLK
1-311

C 14
16-Sep-1984 01:37:58   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:56:51   [BASRTL.SRC]BPAGETBLK.B32;1

Page 5
(3)

```
                                          .PSECT  _BPA$CODE,NOWRT,  SHR,  PIC,2

                         007C 00000       .ENTRY  BPA$GET_BLOCK, Save R2,R3,R4,R5,R6    ; 0207
                   08  AC  DD 00002        PUSHL   BLOCK                                 ; 0250
                   04  AC  9F 00005        PUSHAB  BLOCK_SIZE
         00000000G  00  02  FB 00008       CALLS   #2, LIB$GET_VM
                        56  50 D0 0000F    MOVL    R0, STS
                        0B  56 E9 00012    BLBC    STS, 1$
                   50  08  BC D0 00015     MOVL    @BLOCK, R0                            ; 0252
    04   AC        00  6E  00 2C 00019     MOVC5   #0, (SP), #0, BLOCK_SIZE, (R0)
                        60     0001F
                        50  56 D0 00020 1$: MOVL   STS, R0                              ; 0255
                            04 00023       RET                                          ; 0256
```

; Routine Size: 36 bytes,   Routine Base: _BPA$CODE + 0000

; 165   0257 1
; 166   0258 1 !
; 167   0259 1

BPA$GETFREBLK
1-311

D 14
16-Sep-1984 01:37:58   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:56:51   [BASRTL.SRC]BPAGETBLK.B32;1

Page   6
(4)

```
169    0260  1  GLOBAL ROUTINE bpa$free_block (address, block_size) =    ! deallocate a block
170    0261  1
171    0262  1  !++
172    0263  1  ! FUNCTIONAL DESCRIPTION:
173    0264  1  !
174    0265  1  !     A block 'block_size' bytes long, starting at
175    0266  1  !     address 'address' is returned to the heap storage.
176    0267  1  !     The block must have been allocated using bpa$get_block.
177    0268  1  !
178    0269  1  ! FORMAL PARAMETERS:
179    0270  1  !
180    0271  1  !     address - pointer to block to deallocate
181    0272  1  !     block_size - length of block to deallocate, in bytes
182    0273  1  !
183    0274  1  ! IMPLICIT INPUTS:
184    0275  1  !
185    0276  1  !     NONE
186    0277  1  !
187    0278  1  ! IMPLICIT OUTPUTS:
188    0279  1  !
189    0280  1  !     NONE
190    0281  1  !
191    0282  1  ! ROUTINE VALUE:
192    0283  1  !
193    0284  1  !     success - 1,
194    0285  1  !     failure - ??
195    0286  1  !
196    0287  1  ! SIDE EFFECTS:
197    0288  1  !
198    0289  1  !     A block of heap storage is deallocated.
199    0290  1  !
200    0291  1  !--
201    0292  1
202    0293  2      BEGIN
203    0294  2      RETURN lib$free_vm (block_size, address);     ! M 309
204    0295  1      END;                                          !End of bpa$free_block
```

```
                              0000 00000      .ENTRY   BPA$FREE_BLOCK, Save nothing    ; 0260
                     04  AC 9F 00002          PUSHAB   ADDRESS                         ; 0294
                     08  AC 9F 00005          PUSHAB   BLOCK_SIZE
             00000000G 00   02 FB 00008       CALLS    #2, LIB$FREE_VM
                              04 0000F         RET                                     ; 0295
```

; Routine Size: 16 bytes,   Routine Base: _BPA$CODE + 0024

```
: 205    0296  1 !
: 206    0297  1 !
: 207    0298  1 !<blf/page>
```

BPA$GETFREBLK
1-311

E 14
16-Sep-1984 01:37:58   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:56:51   [BASRTL.SRC]BPAGETBLK.B32;1

Page 7
(5)

```
; 209        0299 1 END                              !End of module
; 210        0300 1
; 211        0301 0 ELUDOM
```

;   PSECT SUMMARY
;
;
;   Name   Bytes   Attributes
;
; _BPA$CODE   52 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

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

; Size: 52 code + 0 data bytes
; Run Time: 00:02.4
; Elapsed Time: 00:07.6
; Lines/CPU Min: 7685
; Lexemes/CPU-Min: 18076
; Memory Used: 24 pages
; Compilation Complete

BASXLATE
LIS

BPAMESAG
LIS

BPAF5S
LIS

BPAGETBLK
LIS

BASZIRE
LIS