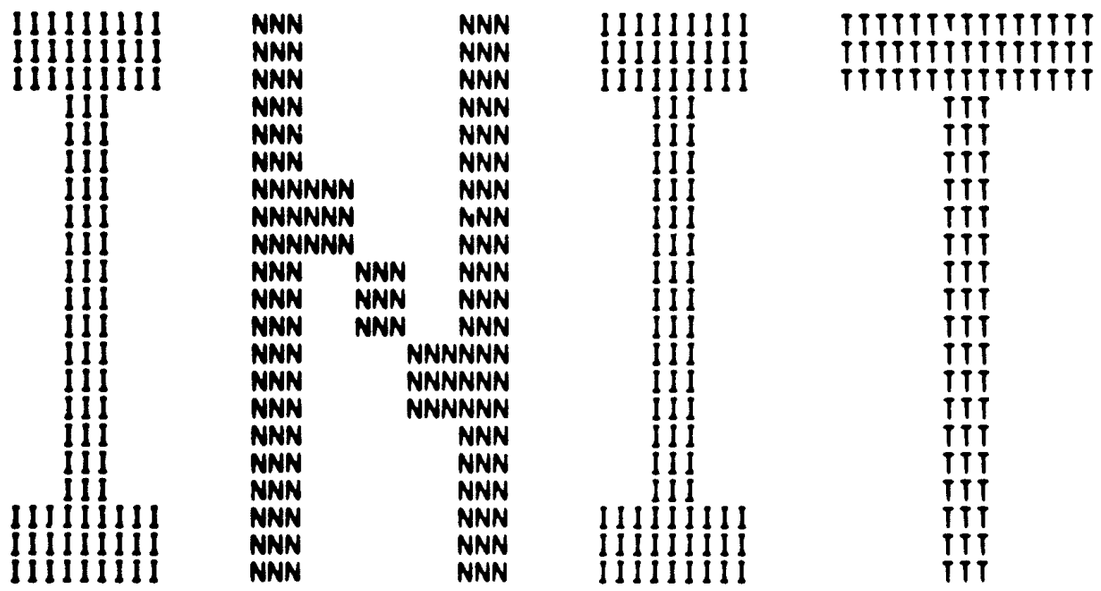
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNNNNN NNN III TTT
III NNNNNN NNN III TTT
III NNNNNN NNN III TTT
III NNN NNN NNN III TTT
III NNN NNN NNN III TTT
III NNN NNN III TTT
III NNN NNNNNN III TTT
III NNN NNNNNN III TTT
III NNN NNNNNN III TTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNN NNN III TTT
IIIIIIIII NNN NNN IIIIIIIII TTT
IIIIIIIII NNN NNN IIIIIIIII TTT
IIIIIIIII NNN NNN IIIIIIIII TTT

FILEID**INIALL

```
IIIIII    NN      NN    IIIIII      AAAAAA    LL          LL
IIIIII    NN      NN    IIIIII      AAAAAA    LL          LL
  II      NN      NN      II      AA      AA  LL          LL
  II      NN      NN      II      AA      AA  LL          LL
  II      NNNN    NN      II      AA      AA  LL          LL
  II      NNNN    NN      II      AA      AA  LL          LL
  II      NN  NN  NN      II      AA      AA  LL          LL
  II      NN  NN  NN      II      AA      AA  LL          LL
  II      NN    NNNN      II      AAAAAAAAAA  LL          LL
  II      NN    NNNN      II      AAAAAAAAAA  LL          LL
  II      NN      NN      II      AA      AA  LL          LL              ....
  II      NN      NN      II      AA      AA  LL          LL              ....
IIIIII    NN      NN    IIIIII    AA      AA  LLLLLLLLLL  LLLLLLLLLL      ....
IIIIII    NN      NN    IIIIII    AA      AA  LLLLLLLLLL  LLLLLLLLLL      ....


LL            IIIIII      SSSSSSS
LL            IIIIII      SSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSS
LL              II        SSSSS
LL              II             SS
LL              II             SS
LL              II             SS
LL              II             SS
LLLLLLLLLL    IIIIII    SSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSS
```

```
;     1          0001  0 MODULE INIALL (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**********************************************************************
;     9          0009  1 !*                                                                    *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                             *
;    13          0013  1 !*                                                                    *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                       *
;    20          0020  1 !*                                                                    *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                       *
;    24          0024  1 !*                                                                    *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;    27          0027  1 !*                                                                    *
;    28          0028  1 !*                                                                    *
;    29          0029  1 !**********************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  INIT Utility Structure Level 1
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This module contains the routines that allocate the pieces of the
;    38          0038  1 !       file structure on the disk.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       STARLET operating system, including privileged system services
;    43          0043  1 !       and internal exec routines.
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 !
;    48          0048  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  12-Nov-1977  20:02
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V02-004 ACG0191         Andrew C. Goldstein,    18-Feb-1981  19:49
;    53          0053  1 !               Fix index file allocation at end of volume
;    54          0054  1 !
;    55          0055  1 !       V0102   ACG0152         Andrew C. Goldstein,    29-Feb-1980  17:01
;    56          0056  1 !               Fix home block delta to correct blocking factor
;    57           057  1 !
```

```
;    58          0058  1 !       V0101   ACG0069         Andrew C. Goldstein,    9-Oct-1979  16:37
;    59          0059  1 !               Remove device data table
;    60          0060  1 !
;    61          0061  1 !       V0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  21:27
;    62          0062  1 !       Previous revision history moved to [INIT.SRC]INIT.REV
;    63          0063  1 !--
;    64          0064  1
;    65          0065  1
;    66          0066  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    67          0067  1 REQUIRE 'SRC$:INIDEF.B32';
;    68          0358  1 REQUIRE 'LIBD$:[VMSLIB.OBJ]INITMSG.B32';
;    69          0490  1
;    70          0491  1
;    71          0492  1 FORWARD ROUTINE
;    72          0493  1         INIT_ALLOCATE   : NOVALUE,      ! main allocation routine
;    73          0494  1         ALLOCATE        : NOVALUE,      ! general allocation scan
;    74          0495  1         ALLOCATE_HOME   : NOVALUE,      ! allocate on home block sequence
;    75          0496  1         CHECK_ALLOC;                    ! verify a candidate allocation
;    76          0497  1
;    77          0498  1
;    78          0499  1
;    79          0500  1 !+
;    80          0501  1 !
;    81          0502  1 ! Module own storage
;    82          0503  1 !
;    83          0504  1 !-
;    84          0505  1
;    85          0506  1 OWN
;    86          0507  1         HOMEBLOCK_DELTA;                ! home block search increment on this volume
```

```
;    88          0508  1 GLOBAL ROUTINE INIT_ALLOCATE : NOVALUE =
;    89          0509  1
;    90          0510  1 !++
;    91          0511  1 !
;    92          0512  1 ! FUNCTIONAL DESCRIPTION:
;    93          0513  1 !
;    94          0514  1 !       This is the main allocation routine. It determines the size and
;    95          0515  1 !       location of each portion of the file structure. Each allocation is
;    96          0516  1 !       done by choosing a candidate location for the section and checking
;    97          0517  1 !       for conflicts. If a conflict exists, a new candidate location is
;    98          0518  1 !       chosen according an algorithm specific to the section being allocated.
;    99          0519  1 !
;   100          0520  1 !
;   101          0521  1 ! CALLING SEQUENCE:
;   102          0522  1 !       INIT_ALLOCATE ()
;   103          0523  1 !
;   104          0524  1 ! INPUT PARAMETERS:
;   105          0525  1 !       NONE
;   106          0526  1 !
;   107          0527  1 ! IMPLICIT INPUTS:
;   108          0528  1 !       parser database
;   109          0529  1 !       allocation table in INIDSK
;   110          0530  1 !
;   111          0531  1 ! OUTPUT PARAMETERS:
;   112          0532  1 !       NONE
;   113          0533  1 !
;   114          0534  1 ! IMPLICIT OUTPUTS:
;   115          0535  1 !       NONE
;   116          0536  1 !
;   117          0537  1 ! ROUTINE VALUE:
;   118          0538  1 !       NONE
;   119          0539  1 !
;   120          0540  1 ! SIDE EFFECTS:
;   121          0541  1 !       allocation table modified
;   122          0542  1 !
;   123          0543  1 !--
;   124          0544  1
;   125          0545  2 BEGIN
;   126          0546  2
;   127          0547  2 EXTERNAL
;   128          0548  2         INIT_OPTIONS    : BITVECTOR,    ! command options
;   129          0549  2         CLUSTER,                        ! volume cluster factor
;   130          0550  2         INDEX,                          ! requested LBN of initial index file
;   131          0551  2         HEADERS,                        ! initial number of file headers
;   132          0552  2         MAXIMUM,                        ! maximum number of files
;   133          0553  2         DIRECTORIES,                    ! number of MFD entries to allocate
;   134          0554  2         VOLUME_SIZE,                    ! size of volume rounded to next cluster
;   135          0555  2         DEVICE_CHAR     : BBLOCK,       ! device charactreistics buffer
;   136          0556  2         BOOTBLOCK_CNT,                  ! block count of boot block cluster
;   137          0557  2         BOOTBLOCK_LBN,                  ! LBN of boot block cluster
;   138          0558  2         HOMEBLOCK1_CNT,                 ! block count of home block 1 cluster
;   139          0559  2         HOMEBLOCK1_LBN,                 ! LBN of home block 1 cluster
;   140          0560  2         HOMEBLOCK2_CNT,                 ! block count of home block 2 cluster
;   141          0561  2         HOMEBLOCK2_LBN,                 ! LBN of home block 2 cluster
;   142          0562  2         IDXFILE_CNT,                    ! block count of initial index file
;   143          0563  2         IDXFILE_LBN,                    ! LBN of initial index file
;   144          0564  2         IDXHDR2_CNT,                    ! block count of 2nd index header cluster
```

```
;  145          0565  2          IDXHDR2_LBN,                    ! LBN of 2nd index header cluster
;  146          0566  2          BITMAP_CNT,                     ! block count of storage bitmap
;  147          0567  2          BITMAP_LBN,                     ! LBN of storage bitmap
;  148          0568  2          MFD_CNT,                        ! block count of MFD
;  149          0569  2          MFD_LBN,                        ! LBN of MFD
;  150          0570  2          VOLEND_CNT,                     ! volume end allocation table entry - count
;  151          0571  2          VOLEND_LBN;                     ! volume end allocation table entry - LBN
;  152          0572  2
;  153          0573  2 EXTERNAL LITERAL
;  154          0574  2          BOOTBLOCK_IDX    : UNSIGNED (6), ! table index of boot block cluster
;  155          0575  2          HOMEBLOCK1_IDX   : UNSIGNED (6), ! table index of home block 1 cluster
;  156          0576  2          HOMEBLOCK2_IDX   : UNSIGNED (6), ! table index of home block 2 cluster
;  157          0577  2          IDXFILE_IDX      : UNSIGNED (6), ! table index of initial index file
;  158          0578  2          IDXHDR2_IDX      : UNSIGNED (6), ! table index of 2nd index header cluster
;  159          0579  2          BITMAP_IDX       : UNSIGNED (6), ! table index of storage bitmap
;  160          0580  2          MFD_IDX          : UNSIGNED (6); ! table index of MFD
;  161          0581  2
;  162          0582  2
;  163          0583  2 ! First make up an allocation pointer to represent the space from the end
;  164          0584  2 ! of the volume to the next 4096 cluster boundary (being the end of the
;  165          0585  2 ! storage map block).
;  166          0586  2 !
;  167          0587  2
;  168          0588  2 VOLEND_CNT = (4096 - (.DEVICE_CHAR[DIB$L_MAXBLOCK] / .CLUSTER) MOD 4096) * .CLUSTER;
;  169          0589  2 VOLEND_LBN = .DEVICE_CHAR[DIB$L_MAXBLOCK] / .CLUSTER * .CLUSTER;
;  170          0590  2
;  171          0591  2 ! Allocate the boot block to the first available cluster (usually 0).
;  172          0592  2 !
;  173          0593  2
;  174          0594  2 BOOTBLOCK_CNT = 1;
;  175          0595  2 ALLOCATE (BOOTBLOCK_IDX, 0);
;  176          0596  2
;  177          0597  2 IF .BOOTBLOCK_LBN NEQ 0
;  178          0598  2 THEN ERR_MESSAGE (INIT$_BLKZERO);
;  179          0599  2
;  180          0600  2 ! Next allocate the primary and secondary home blocks. If the boot block is
;  181          0601  2 ! on LBN 0 and the cluster factor is greater than 1, then the primary home
;  182          0602  2 ! block cluster is a dummy since the real home block is LBN 1.
;  183          0603  2 !
;  184          0604  2
;  185          0605  2 HOMEBLOCK1_CNT = 1;
;  186          0606  2
;  187          0607  2 IF .INIT_OPTIONS[OPT_STRUCTURE1]
;  188          0608  2 THEN
;  189          0609  2     ALLOCATE_HOME (HOMEBLOCK1_IDX)
;  190          0610  2 ELSE
;  191          0611  3     BEGIN
;  192          0612  3     IF .BOOTBLOCK_LBN EQL 0 AND .CLUSTER GTR 1
;  193          0613  3     THEN
;  194          0614  3         ALLOCATE (HOMEBLOCK1_IDX, 0)
;  195          0615  3     ELSE
;  196          0616  3         ALLOCATE_HOME (HOMEBLOCK1_IDX);
;  197          0617  3
;  198          0618  3     HOMEBLOCK2_CNT = 1;
;  199          0619  3     ALLOCATE_HOME (HOMEBLOCK2_IDX);
;  200          0620  2     END;
;  201          0621  2
```

```
;  202          0622  2 ! Now allocate the MFD, storage map, initial index file, and alternate
;  203          0623  2 ! index file header, in that order. This results in optimal locality of
;  204          0624  2 ! the most frequently referenced portions of the file structure. Note that
;  205          0625  2 ! if the index file is being placed at the end of the volume they are
;  206          0626  2 ! allocated in reverse to achieve the same effect.
;  207          0627  2 !
;  208          0628  2
;  209          0629  2 MFD_LBN = .INDEX;
;  210          0630  2 BITMAP_LBN = .INDEX;
;  211          0631  2 IDXFILE_LBN = .INDEX;
;  212          0632  2
;  213          0633  2 IF NOT .INIT_OPTIONS[OPT_INDEX_END]
;  214          0634  2 THEN
;  215          0635  3     BEGIN
;  216          0636  3     MFD_CNT = .DIRECTORIES/16 + 1;
;  217          0637  3     ALLOCATE (MFD_IDX, 0);
;  218          0638  3     BITMAP_CNT = ((.VOLUME_SIZE/.CLUSTER + 4095) / 4096) + 1;
;  219          0639  3     ALLOCATE (BITMAP_IDX, 0);
;  220          0640  3     IDXFILE_CNT = .HEADERS + (.MAXIMUM+4095)/4096;
;  221          0641  3     ALLOCATE (IDXFILE_IDX, 0);
;  222          0642  3     END
;  223          0643  2 ELSE
;  224          0644  3     BEGIN
;  225          0645  3     IDXFILE_CNT = .HEADERS + (.MAXIMUM+4095)/4096;
;  226          0646  3     ALLOCATE (IDXFILE_IDX, 1);
;  227          0647  3     BITMAP_CNT = ((.VOLUME_SIZE/.CLUSTER + 4095) / 4096) + 1;
;  228          0648  3     ALLOCATE (BITMAP_IDX, 1);
;  229          0649  3     MFD_CNT = .DIRECTORIES/16 + 1;
;  230          0650  3     ALLOCATE (MFD_IDX, 1);
;  231          0651  2     END;
;  232          0652  2
;  233          0653  2 IF NOT .INIT_OPTIONS[OPT_STRUCTURE1]
;  234          0654  2 THEN
;  235          0655  3     BEGIN
;  236          0656  3     IDXHDR2_CNT = 1;
;  237          0657  3     IDXHDR2_LBN = .IDXFILE_LBN + .HOMEBLOCK_DELTA;
;  238          0658  3     IF .INIT_OPTIONS[OPT_INDEX_END]
;  239          0659  3     THEN
;  240          0660  4         BEGIN
;  241          0661  4         IDXHDR2_LBN = .IDXFILE_LBN - .HOMEBLOCK_DELTA;
;  242          0662  4         ALLOCATE (IDXHDR2_IDX, 1);
;  243          0663  4         END
;  244          0664  3     ELSE
;  245          0665  3         ALLOCATE (IDXHDR2_IDX, 0);
;  246          0666  2     END;
;  247          0667  2
;  248          0668  1 END;                                    ! end of routine INIT_ALLOCATE


                                                                .TITLE  INIALL
                                                                .IDENT  \V04-000\

                                                                .PSECT  $OWN$,NOEXE,2

                                                    00000 HOMEBLOCK_DELTA:
                                                                .BLKB   4
```

```
                                        .EXTRN  INIT_OPTIONS, CLUSTER
                                        .EXTRN  INDEX, HEADERS, MAXIMUM
                                        .EXTRN  DIRECTORIES, VOLUME_SIZE
                                        .EXTRN  DEVICE_CHAR, BOOTBLOCK_CNT
                                        .EXTRN  BOOTBLOCK_LBN, HOMEBLOCK1_CNT
                                        .EXTRN  HOMEBLOCK1_LBN, HOMEBLOCK2_CNT
                                        .EXTRN  HOMEBLOCK2_LBN, IDXFILE_CNT
                                        .EXTRN  IDXFILE_LBN, IDXHDR2_CNT
                                        .EXTRN  IDXHDR2_LBN, BITMAP_CNT
                                        .EXTRN  BITMAP_LBN, MFD_CNT
                                        .EXTRN  MFD_LBN, VOLEND_CNT
                                        .EXTRN  VOLEND_LBN, BOOTBLOCK_IDX
                                        .EXTRN  HOMEBLOCK1_IDX, HOMEBLOCK2_IDX
                                        .EXTRN  IDXFILE_IDX, IDXHDR2_IDX
                                        .EXTRN  BITMAP_IDX, MFD_IDX

                                        .PSECT  $CODE$,NOWRT,2

                                003C 00000      .ENTRY  INIT_ALLOCATE, Save R2,R3,R4,R5                 ; 0508
                            55  0000G  CF   9E 00002  MOVAB   IDXFILE_LBN, R5
                            54  0000G  CF   9E 00007  MOVAB   CLUSTER, R4
                            53  0000V  CF   9E 0000C  MOVAB   ALLOCATE, R3
                            51        64   D0 00011  MOVL    CLUSTER, R1                               ; 0588
                     52     0000G     CF   51   C7 00014  DIVL3   R1, DEVICE_CHAR+112, R2
                7E   00               52   01   7A 0001A  EMUL    #1, R2, #0, -(SP)
                50   50          8E 00001000   8F   7B 0001F  EDIV    #4096, (SP)+, R0, R0
                                    50   F000   C0   9E 00028  MOVAB   -4096(R0), R0
                                    50         51   C4 0002D  MULL2   R1, R0
                            0000G   CF         50   CE 00030  MNEGL   R0, VOLEND_CNT
                     0000G  CF               52   51   C5 00035  MULL3   R1, R2, VOLEND_LBN             ; 0589
                            0000G   CF         01   D0 0003B  MOVL    #1, BOOTBLOCK_CNT                 ; 0594
                                               7E   D4 00040  CLRL    -(SP)                             ; 0595
                                    7E         00G  9A 00042  MOVZBL  S^BOOTBLOCK_IDX, -(SP)
                                    63         02   FB 00045  CALLS   #2, ALLOCATE
                                          0000G   CF   D5 00048  TSTL    BOOTBLOCK_LBN                   ; 0597
                                                0D   13 0004C  BEQL    1$
                                      00759000    8F   DD 0004E  PUSHL   #7704576                        ; 0598
                              00000000G   00      01   FB 00054  CALLS   #1, LIB$SIGNAL
                                0000G     CF      01   D0 0005B  1$:    MOVL    #1, HOMEBLOCK1_CNT      ; 0605
                                          0000G   CF   95 00060  TSTB    INIT_OPTIONS+3                  ; 0607
                                                05   18 00064  BGEQ    2$
                                    7E         00G  9A 00066  MOVZBL  S^HOMEBLOCK1_IDX, -(SP)           ; 0609
                                               25   11 00069  BRB     5$
                                          0000G   CF   D5 0006B  2$:    TSTL    BOOTBLOCK_LBN           ; 0612
                                                0F   12 0006F  BNEQ    3$
                                    01         64   D1 00071  CMPL    CLUSTER, #1
                                               0A   15 00074  BLEQ    3$
                                               7E   D4 00076  CLRL    -(SP)                           ; 0614
                                    7E         00G  9A 00078  MOVZBL  S^HOMEBLOCK1_IDX, -(SP)
                                    63         02   FB 0007B  CALLS   #2, ALLOCATE
                                               08   11 0007E  BRB     4$
                                    7E         00G  9A 00080  3$:    MOVZBL  S^HOMEBLOCK1_IDX, -(SP)   ; 0616
                              0000V CF         01   FB 00083  CALLS   #1, ALLOCATE_HOME
                              0000G CF         01   D0 00088  4$:    MOVL    #1, HOMEBLOCK2_CNT        ; 0618
                                    7E         00G  9A 0008D  MOVZBL  S^HOMEBLOCK2_IDX, -(SP)           ; 0619
                              0000V CF         01   FB 00090  5$:    CALLS   #1, ALLOCATE_HOME
                                    50   0000G   CF   D0 00095  MOVL    INDEX, R0                       ; 0629
```

```
                 0000G  CF              50  D0 0009A         MOVL    R0, MFD_LBN                                  ;
                 0000G  CF              50  D0 0009F         MOVL    R0, BITMAP_LBN                               ; 0630
                        65              50  D0 000A4         MOVL    R0, IDXFILE_LBN                              ; 0631
          54     0000G  CF              06  E0 000A7         BBS     #6, INIT_OPTIONS+2, 6$                       ; 0633
          50     0000G  CF              10  C7 000AD         DIVL3   #16, DIRECTORIES, R0                         ; 0636
                 0000G  CF        01    A0  9E 000B3         MOVAB   1(R0), MFD_CNT                               ;
                                        7E  D4 000B9         CLRL    -(SP)                                        ; 0637
                        7E             00G  9A 000BB         MOVZBL  S^MFD_IDX, -(SP)                             ;
                        63              02  FB 000BE         CALLS   #2, ALLOCATE                                 ;
          50     0000G  CF              64  C7 000C1         DIVL3   CLUSTER, VOLUME_SIZE, R0                     ; 0638
                        50      0FFF    C0  9E 000C7         MOVAB   4095(R0), R0                                 ;
                        50 00001000     8F  C6 000CC         DIVL2   #4096, R0                                    ;
                 0000G  CF        01    A0  9E 000D3         MOVAB   1(R0), BITMAP_CNT                            ;
                                        7E  D4 000D9         CLRL    -(SP)                                        ; 0639
                        7E             00G  9A 000DB         MOVZBL  S^BITMAP_IDX, -(SP)                          ;
                        63              02  FB 000DE         CALLS   #2, ALLOCATE                                 ;
          50     0000G  CF 00000FFF     8F  C1 000E1         ADDL3   #4095, MAXIMUM, R0                           ; 0640
                        50 00001000     8F  C6 000EB         DIVL2   #4096, R0                                    ;
                 0000G  CF      0000GDF40   9E 000F2         MOVAB   @HEADERS[R0], IDXFILE_CNT                    ;
                                        7E  D4 000FA         CLRL    -(SP)                                        ; 0641
                        7E             00G  9A 000FC         MOVZBL  S^IDXFILE_IDX, -(SP)                         ;
                                        52  11 000FF         BRB     7$                                           ;
          50     0000G  CF 00000FFF     8F  C1 00101 6$:     ADDL3   #4095, MAXIMUM, R0                           ; 0645
                        50 00001000     8F  C6 0010B         DIVL2   #4096, R0                                    ;
                 0000G  CF      0000GDF40   9E 00112         MOVAB   @HEADERS[R0], IDXFILE_CNT                    ;
                                        01  DD 0011A         PUSHL   #1                                           ; 0646
                        7E             00G  9A 0011C         MOVZBL  S^IDXFILE_IDX, -(SP)                         ;
                        63              02  FB 0011F         CALLS   #2, ALLOCATE                                 ;
          50     0000G  CF              64  C7 00122         DIVL3   CLUSTER, VOLUME_SIZE, R0                     ; 0647
                        50      0FFF    C0  9E 00128         MOVAB   4095(R0), R0                                 ;
                        50 00001000     8F  C6 0012D         DIVL2   #4096, R0                                    ;
                 0000G  CF        01    A0  9E 00134         MOVAB   1(R0), BITMAP_CNT                            ;
                                        01  DD 0013A         PUSHL   #1                                           ; 0648
                        7E             00G  9A 0013C         MOVZBL  S^BITMAP_IDX, -(SP)                          ;
                        63              02  FB 0013F         CALLS   #2, ALLOCATE                                 ;
          50     0000G  CF              10  C7 00142         DIVL3   #16, DIRECTORIES, R0                         ; 0649
                 0000G  CF        01    A0  9E 00148         MOVAB   1(R0), MFD_CNT                               ;
                                        01  DD 0014E         PUSHL   #1                                           ; 0650
                        7E             00G  9A 00150         MOVZBL  S^MFD_IDX, -(SP)                             ;
                        63              02  FB 00153 7$:     CALLS   #2, ALLOCATE                                 ;
                              0000G     CF  95 00156         TSTB    INIT_OPTIONS+3                               ; 0653
                                        27  19 0015A         BLSS    10$                                          ;
                 0000G  CF              01  D0 0015C         MOVL    #1, IDXHDR2_CNT                              ; 0656
     0000G  CF          65    0000'     CF  C1 00161         ADDL3   HOMEBLOCK_DELTA, IDXFILE_LBN, IDXHDR2_LBN    ; 0657
          0C     0000G  CF              06  E1 00169         BBC     #6, INIT_OPTIONS+2, 8$                       ; 0658
     0000G  CF          65    0000'     CF  C3 0016F         SUBL3   HOMEBLOCK_DELTA, IDXFILE_LBN, IDXHDR2_LBN    ; 0661
                                        01  DD 00177         PUSHL   #1                                           ; 0662
                                        02  11 00179         BRB     9$                                           ;
                                        7E  D4 0017B 8$:     CLRL    -(SP)                                        ; 0665
                        7E             00G  9A 0017D 9$:     MOVZBL  S^IDXHDR2_IDX, -(SP)                         ;
                        63              02  FB 00180         CALLS   #2, ALLOCATE                                 ;
                                            04 00183 10$:    RET                                                  ; 0668

; Routine Size:  388 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  250     0669  1 ROUTINE ALLOCATE (INDEX, REVERSE) : NOVALUE =
;  251     0670  1
;  252     0671  1 !++
;  253     0672  1 !
;  254     0673  1 ! FUNCTIONAL DESCRIPTION:
;  255     0674  1 !
;  256     0675  1 !       This routine allocates the given table entry in the first available
;  257     0676  1 !       position after the given start, searching in the given direction.
;  258     0677  1 !
;  259     0678  1 !
;  260     0679  1 ! CALLING SEQUENCE:
;  261     0680  1 !       ALLOCATE (ARG1, ARG2)
;  262     0681  1 !
;  263     0682  1 ! INPUT PARAMETERS:
;  264     0683  1 !       ARG1: allocation table index of entry to allocate
;  265     0684  1 !       ARG2: direction: 0 = forward
;  266     0685  1 !                        1 = reverse
;  267     0686  1 !
;  268     0687  1 ! IMPLICIT INPUTS:
;  269     0688  1 !       allocation table
;  270     0689  1 !
;  271     0690  1 ! OUTPUT PARAMETERS:
;  272     0691  1 !       NONE
;  273     0692  1 !
;  274     0693  1 ! IMPLICIT OUTPUTS:
;  275     0694  1 !       entry in allocation table
;  276     0695  1 !
;  277     0696  1 ! ROUTINE VALUE:
;  278     0697  1 !       NONE
;  279     0698  1 !
;  280     0699  1 ! SIDE EFFECTS:
;  281     0700  1 !       NONE
;  282     0701  1 !
;  283     0702  1 !--
;  284     0703  1
;  285     0704  2 BEGIN
;  286     0705  2
;  287     0706  2 LOCAL
;  288     0707  2         CONFLICT;                       ! index of conflicting table entry
;  289     0708  2
;  290     0709  2 EXTERNAL
;  291     0710  2         CLUSTER,                        ! volume cluster factor
;  292     0711  2         VOLUME_SIZE,                    ! size of volume rounded to next cluster
;  293     0712  2         ALLOC_TABLE_CNT : VECTOR,       ! allocation count table
;  294     0713  2         ALLOC_TABLE_LBN : VECTOR;       ! allocation LBN table
;  295     0714  2
;  296     0715  2
;  297     0716  2 ! Round the starting LBN and count down and up, respectively, to cluster boundaries.
;  298     0717  2 ! Iterate, checking the proposed location of the entry against the rest of
;  299     0718  2 ! the allocation table. When we encounter a conflict, adjust the location
;  300     0719  2 ! past the conflicting entry and try again.
;  301     0720  2 !
;  302     0721  2
;  303     0722  2 ALLOC_TABLE_LBN[.INDEX] = .ALLOC_TABLE_LBN[.INDEX] / .CLUSTER * .CLUSTER;
;  304     0723  2 ALLOC_TABLE_CNT[.INDEX] = (.ALLOC_TABLE_CNT[.INDEX] + .CLUSTER - 1) / .CLUSTER * .CLUSTER;
;  305     0724  2 WHILE 1 DO
;  306     0725  3     BEGIN
```

```
0726 3
0727 3 ! The limit test works in the reverse direction since we will wrap through
0728 3 ! zero.
0729 3 !
0730 3
0731 3     IF .ALLOC_TABLE_LBN[.INDEX] GEQU .VOLUME_SIZE
0732 3     THEN ERR_EXIT (INIT$_ALLOCFAIL);
0733 3
0734 3     CONFLICT = CHECK_ALLOC (.INDEX);
0735 3     IF .CONFLICT EQL -1 THEN RETURN;
0736 3
0737 3     IF NOT .REVERSE
0738 3     THEN                                  ! search in forward direction
0739 3         ALLOC_TABLE_LBN[.INDEX] = .ALLOC_TABLE_LBN[.CONFLICT]
0740 3                                 + .ALLOC_TABLE_CNT[.CONFLICT]
0741 3
0742 3     ELSE                                  ! search in reverse direction
0743 3         ALLOC_TABLE_LBN[.INDEX] = .ALLOC_TABLE_LBN[.CONFLICT]
0744 3                                 - .ALLOC_TABLE_CNT[.INDEX];
0745 2     END;
0746 2
0747 1 END;                                      ! end of routine ALLOCATE


                                                         .EXTRN  ALLOC_TABLE_CNT
                                                         .EXTRN  ALLOC_TABLE_LBN

                                   00FC 00000 ALLOCATE:
                                                         .WORD   Save R2,R3,R4,R5,R6,R7                     ; 0669
                57      0000G  CF   9E 00002             MOVAB   ALLOC_TABLE_LBN, R7                        ;
                53         04  AC   D0 00007             MOVL    INDEX, R3                                  ; 0722
                54         6743     DE 0000B             MOVAL   ALLOC_TABLE_LBN[R3], R4                    ;
                51      0000G  CF   D0 0000F             MOVL    CLUSTER, R1                                ;
         50     64             51   C7 00014             DIVL3   R1, (R4), R0                               ;
         64     50             51   C5 00018             MULL3   R1, R0, (R4)                               ;
                55      0000GCF43   DE 0001C             MOVAL   ALLOC_TABLE_CNT[R3], R5                    ; 0723
         50     65             51   C1 00022             ADDL3   R1, (R5), R0                               ;
                               50   D7 00026             DECL    R0                                         ;
                50             51   C6 00028             DIVL2   R1, R0                                     ;
         65     50             51   C5 0002B             MULL3   R1, R0, (R5)                               ;
                56         08  AC   D2 0002F             MCOML   REVERSE, R6                                ; 0737
          0000G CF             64   D1 00033 1$:         CMPL    (R4), VOLUME_SIZE                          ; 0731
                               0D   1F 00038             BLSSU   2$                                         ;
                     0075807C  8F   DD 0003A             PUSHL   #7700604                                   ; 0732
      000000000G 00            01   FB 00040             CALLS   #1, LIB$STOP                               ;
                               53   DD 00047 2$:         PUSHL   R3                                         ; 0734
          0000V CF             01   FB 00049             CALLS   #1, CHECK_ALLOC                            ;
                52             50   D0 0004E             MOVL    R0, CONFLICT                               ;
       FFFFFFFF 8F             52   D1 00051             CMPL    CONFLICT, #-1                              ; 0735
                               14   13 00058             BEQL    4$                                         ;
                0A             56   E9 0005A             BLBC    R6, 3$                                     ; 0739
         64         6742  0000GCF42 C1 0005D             ADDL3   ALLOC_TABLE_CNT[CONFLICT], ALLOC_TABLE_LBN- ; 0740
                                                                 [CONFLICT], (R4)
                               CC   11 00065             BRB     1$                                         ; 0739
         64         6742       65   C3 00067 3$:         SUBL3   (R5), ALLOC_TABLE_LBN[CONFLICT], (R4)      ; 0744
                               C5   11 0006C             BRB     1$                                         ; 0724
```

```
                                04 0006E 4$:      RET                                             ; 0747

; Routine Size:  111 bytes,    Routine Base:  $CODE$ + 0184
```

```
;  330     0748  1 ROUTINE ALLOCATE_HOME (INDEX) : NOVALUE =
;  331     0749  1
;  332     0750  1 !++
;  333     0751  1 !
;  334     0752  1 ! FUNCTIONAL DESCRIPTION:
;  335     0753  1 !
;  336     0754  1 !       This routine allocates the indicated allocation table entry to
;  337     0755  1 !       the first available block on the home block search sequence.
;  338     0756  1 !
;  339     0757  1 !
;  340     0758  1 ! CALLING SEQUENCE:
;  341     0759  1 !       ALLOC_HOME (ARG1)
;  342     0760  1 !
;  343     0761  1 ! INPUT PARAMETERS:
;  344     0762  1 !       ARG1: table index of home block cluster
;  345     0763  1 !
;  346     0764  1 ! IMPLICIT INPUTS:
;  347     0765  1 !       allocation table in INIDSK
;  348     0766  1 !
;  349     0767  1 ! OUTPUT PARAMETERS:
;  350     0768  1 !       NONE
;  351     0769  1 !
;  352     0770  1 ! IMPLICIT OUTPUTS:
;  353     0771  1 !       entry in table
;  354     0772  1 !
;  355     0773  1 ! ROUTINE VALUE:
;  356     0774  1 !       NONE
;  357     0775  1 !
;  358     0776  1 ! SIDE EFFECTS:
;  359     0777  1 !       NONE
;  360     0778  1 !
;  361     0779  1 !--
;  362     0780  1
;  363     0781  2 BEGIN
;  364     0782  2
;  365     0783  2 LOCAL
;  366     0784  2         DELTA,                          ! home block search delta
;  367     0785  2         BLOCKFACT,                      ! device blocking factor
;  368     0786  2         LBN;                            ! home block candidate LBN
;  369     0787  2
;  370     0788  2 EXTERNAL
;  371     0789  2         INIT_OPTIONS    : BITVECTOR,    ! command options
;  372     0790  2         DEVICE_CHAR     : BBLOCK,       ! device characteristics
;  373     0791  2         CLUSTER,                        ! volume cluster factor
;  374     0792  2         VOLUME_SIZE,                    ! size of volume rounded to next cluster
;  375     0793  2         REAL_HOMEBLOCK,                 ! LBN of "official" home block
;  376     0794  2         ALLOC_TABLE_CNT : VECTOR,       ! allocation count table
;  377     0795  2         ALLOC_TABLE_LBN : VECTOR;       ! allocation LBN table
;  378     0796  2
;  379     0797  2
;  380     0798  2 ! Compute the home block search delta. For structure level 1, this is simply
;  381     0799  2 ! 256, except that the first slot is on LBN 1 rather than 0. For level 2,
;  382     0800  2 ! compute the home block search delta from the volume geometry in the
;  383     0801  2 ! device table. This is done according to the following rules, where volume
;  384     0802  2 ! geometry is expressed in the order sectors, tracks, cylinders:
;  385     0803  2 !
;  386     0804  2 !       n x 1 x 1:      1
```

```
;  387          0805  2 !       1 x n x 1:      1
;  388          0806  2 !       1 x 1 x n:      1
;  389          0807  2 !
;  390          0808  2 !       n x m x 1:      n+1
;  391          0809  2 !       n x 1 x m:      n+1
;  392          0810  2 !       1 x n x m:      n+1
;  393          0811  2 !
;  394          0812  2 !       s x t x c:      (t+1)*s+1
;  395          0813  2 !
;  396          0814  2
;  397          0815  2 IF .INIT_OPTIONS[OPT_STRUCTURE1]
;  398          0816  2 THEN
;  399          0817  2     DELTA = 256
;  400          0818  2 ELSE
;  401          0819  3     BEGIN
;  402          0820  4     BLOCKFACT = (.DEVICE_CHAR[DIB$B_SECTORS]
;  403          0821  4                  * .DEVICE_CHAR[DIB$B_TRACKS]
;  404          0822  4                  * .DEVICE_CHAR[DIB$W_CYLINDERS])
;  405          0823  3                  / .DEVICE_CHAR[DIB$L_MAXBLOCK];
;  406          0824  3
;  407          0825  3     DELTA = 1;
;  408          0826  3     IF  .DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1
;  409          0827  3     AND .DEVICE_CHAR[DIB$B_TRACKS] GTR 1
;  410          0828  3     THEN DELTA = .DELTA + .DEVICE_CHAR[DIB$B_TRACKS];
;  411          0829  3
;  412          0830  3     IF  .DEVICE_CHAR[DIB$B_SECTORS] GTR 1
;  413          0831  4     AND (.DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1
;  414          0832  4         OR .DEVICE_CHAR[DIB$B_TRACKS] GTR 1)
;  415          0833  3     THEN DELTA = (.DELTA * .DEVICE_CHAR[DIB$B_SECTORS] + .BLOCKFACT) / .BLOCKFACT;
;  416          0834  3
;  417          0835  3     IF .DELTA EQL 0
;  418          0836  3     OR .DELTA GTRU .DEVICE_CHAR[DIB$L_MAXBLOCK] / 10
;  419          0837  3     THEN DELTA = 1;
;  420          0838  2     END;
;  421          0839  2
;  422          0840  2 HOMEBLOCK_DELTA = .DELTA;
;  423          0841  2
;  424          0842  2 ! Round the starting LBN and count down and up, respectively, to cluster boundaries.
;  425          0843  2 ! Now find the first available cluster on the search sequence by starting
;  426          0844  2 ! with LBN 1 and incrementing by the delta for each try.
;  427          0845  2 !
;  428          0846  2
;  429          0847  2 LBN = 1;
;  430          0848  2
;  431          0849  2 ALLOC_TABLE_CNT[.INDEX] = (.ALLOC_TABLE_CNT[.INDEX] + .CLUSTER - 1) / .CLUSTER * .CLUSTER;
;  432          0850  2 WHILE 1 DO
;  433          0851  3     BEGIN
;  434          0852  3     ALLOC_TABLE_LBN[.INDEX] = .LBN / .CLUSTER * .CLUSTER;
;  435          0853  3     IF CHECK_ALLOC (.INDEX) EQL -1 THEN EXITLOOP;
;  436          0854  3     IF .INIT_OPTIONS[OPT_STRUCTURE1]
;  437          0855  3     THEN LBN = .LBN AND NOT 1;
;  438          0856  3     LBN = .LBN + .DELTA;
;  439          0857  3     IF .LBN GEQU .VOLUME_SIZE
;  440          0858  3     THEN ERR_EXIT (INIT$_ALLOCFAIL);
;  441          0859  2     END;
;  442          0860  2
;  443          0861  2 REAL_HOMEBLOCK = .LBN;                  ! save LBN of actual block
```

```
;  444          0862  2
;  445          0863  1 END;                                  ! end of routine ALLOCATE_HOME


                                                                   .EXTRN  REAL_HOMEBLOCK

                                     007C 00000 ALLOCATE_HOME:
                                                                   .WORD   Save R2,R3,R4,R5,R6                        ; 0748
                       56       0000G  CF   9E 00002               MOVAB   CLUSTER, R6                                ;
                       55       0000G  CF   9E 00007               MOVAB   DEVICE_CHAR+10, R5                         ;
                                0000G  CF   95 0000C               TSTB    INIT_OPTIONS+3                             ; 0815
                                       07   18 00010               BGEQ    1$                                         ;
                       53       0100   8F   3C 00012               MOVZWL  #256, DELTA                                ; 0817
                                       54   11 00017               BRB     6$                                         ;
                       52         FE   A5   9A 00019 1$:           MOVZBL  DEVICE_CHAR+8, R2                          ; 0820
                       51         FF   A5   9A 0001D               MOVZBL  DEVICE_CHAR+9, R1                          ; 0821
              50       52              51   C5 00021               MULL3   R1, R2, R0                                 ;
                       54              65   3C 00025               MOVZWL  DEVICE_CHAR+10, R4                         ; 0822
                       50              54   C4 00028               MULL2   R4, R0                                     ;
              54       50         66   A5   C7 0002B               DIVL3   DEVICE_CHAR+112, R0, BLOCKFACT             ; 0823
                       53              01   D0 00030               MOVL    #1, DELTA                                  ; 0825
                                       50   D4 00033               CLRL    R0                                         ; 0826
                       01              65   B1 00035               CMPW    DEVICE_CHAR+10, #1                         ;
                                       0A   1B 00038               BLEQU   2$                                         ;
                                       50   D6 0003A               INCL    R0                                         ;
                       01              51   91 0003C               CMPB    R1, #1                                     ; 0827
                                       03   1B 0003F               BLEQU   2$                                         ;
                       53              51   C0 00041               ADDL2   R1, DELTA                                  ; 0828
                       01              52   91 00044 2$:           CMPB    R2, #1                                     ; 0830
                                       13   1B 00047               BLEQU   4$                                         ;
                       05              50   E8 00049               BLBS    R0, 3$                                     ; 0831
                       01              51   91 0004C               CMPB    R1, #1                                     ; 0832
                                       0B   1B 0004F               BLEQU   4$                                         ;
              50       53              52   C5 00051 3$:           MULL3   R2, DELTA, R0                              ; 0833
                       50              54   C0 00055               ADDL2   BLOCKFACT, R0                              ;
              53       50              54   C7 00058               DIVL3   BLOCKFACT, R0, DELTA                       ;
                                       53   D5 0005C 4$:           TSTL    DELTA                                      ; 0835
                                       0A   13 0005E               BEQL    5$                                         ;
              50          66   A5      0A   C7 00060               DIVL3   #10, DEVICE_CHAR+112, R0                   ; 0836
                       50              53   D1 00065               CMPL    DELTA, R0                                  ;
                                       03   1B 00068               BLEQU   6$                                         ;
                       53              01   D0 0006A 5$:           MOVL    #1, DELTA                                  ; 0837
                  0000'    CF          53   D0 0006D 6$:           MOVL    DELTA, HOMEBLOCK_DELTA                     ; 0840
                       52              01   D0 00072               MOVL    #1, LBN                                    ; 0847
                       54         04   AC   D0 00075               MOVL    INDEX, R4                                  ; 0849
                       50       0000GCF44   D0 00079               MOVL    ALLOC_TABLE_CNT[R4], R0                    ;
                       51              66   D0 0007F               MOVL    CLUSTER, R1                                ;
                       50         FF A140   9E 00082               MOVAB   -1(R1)[R0], R0                             ;
                       50              51   C6 00087               DIVL2   R1, R0                                     ;
       0000GCF44       50              51   C5 0008A               MULL3   R1, R0, ALLOC_TABLE_CNT[R4]                ;
              50       52              66   C7 00091 7$:           DIVL3   CLUSTER, LBN, R0                           ; 0852
       0000GCF44       50              66   C5 00095               MULL3   CLUSTER, R0, ALLOC_TABLE_LBN[R4]           ;
                                       54   DD 0009C               PUSHL   R4                                         ; 0853
                  0000V    CF          01   FB 0009E               CALLS   #1, CHECK_ALLOC                            ;
               FFFFFFFF    8F          50   D1 000A3               CMPL    R0, #-1                                    ;
                                       22   13 000AA               BEQL    9$                                         ;
```

```
                                 0000G  CF  95 000AC          TSTB    INIT_OPTIONS+3                          ; 0854
                                        03  18 000B0          BGEQ    8$                                      ;
                              52        01  8A 000B2          BICB2   #1, LBN                                 ; 0855
                              52        53  C0 000B5 8$:      ADDL2   DELTA, LBN                              ; 0856
                        0000G CF        52  D1 000B8          CMPL    LBN, VOLUME_SIZE                        ; 0857
                                        D2  1F 000BD          BLSSU   7$                                      ;
                           0075807C     8F  DD 000BF          PUSHL   #7700604                                ; 0858
                  00000000G  00         01  FB 000C5          CALLS   #1, LIB$STOP                            ;
                                        C3  11 000CC          BRB     7$                                      ; 0850
                        0000G CF        52  D0 000CE 9$:      MOVL    LBN, REAL_HOMEBLOCK                     ; 0861
                                            04 000D3          RET                                             ; 0863

; Routine Size:  212 bytes,    Routine Base:  $CODE$ + 01F3
```

```
;  447          0864  1 ROUTINE CHECK_ALLOC (INDEX) =
;  448          0865  1
;  449          0866  1 !++
;  450          0867  1 !
;  451          0868  1 ! FUNCTIONAL DESCRIPTION:
;  452          0869  1 !
;  453          0870  1 !       This routine checks the indicated allocation table entry for
;  454          0871  1 !       conflicts against all other table entries.
;  455          0872  1 !
;  456          0873  1 !
;  457          0874  1 ! CALLING SEQUENCE:
;  458          0875  1 !       CHECK_ALLOC (ARG1)
;  459          0876  1 !
;  460          0877  1 ! INPUT PARAMETERS:
;  461          0878  1 !       ARG1: index of table entry to check
;  462          0879  1 !
;  463          0880  1 ! IMPLICIT INPUTS:
;  464          0881  1 !       allocation table in INIDSK
;  465          0882  1 !
;  466          0883  1 ! OUTPUT PARAMETERS:
;  467          0884  1 !       NONE
;  468          0885  1 !
;  469          0886  1 ! IMPLICIT OUTPUTS:
;  470          0887  1 !       NONE
;  471          0888  1 !
;  472          0889  1 ! ROUTINE VALUE:
;  473          0890  1 !       index of conflicting table entry
;  474          0891  1 !       or -1 if no conflict
;  475          0892  1 !
;  476          0893  1 ! SIDE EFFECTS:
;  477          0894  1 !       NONE
;  478          0895  1 !
;  479          0896  1 !--
;  480          0897  1
;  481          0898  2 BEGIN
;  482          0899  2
;  483          0900  2 EXTERNAL
;  484          0901  2         ALLOC_TABLE_CNT : VECTOR,       ! allocation count table
;  485          0902  2         ALLOC_TABLE_LBN : VECTOR;       ! allocation LBN table
;  486          0903  2
;  487          0904  2 EXTERNAL LITERAL
;  488          0905  2         ALLOC_MAX       : UNSIGNED (16); ! total size of allocation table
;  489          0906  2
;  490          0907  2
;  491          0908  2 ! Simply scan the entire table, doing a range compare on each entry (noting
;  492          0909  2 ! not to compare the candidate against itself). Active table entries are
;  493          0910  2 ! identified by a non-zero count.
;  494          0911  2 !
;  495          0912  2
;  496          0913  2 INCR J FROM 0 TO ALLOC_MAX DO
;  497          0914  3     BEGIN
;  498          0915  3     IF .ALLOC_TABLE_CNT[.J] NEQ 0
;  499          0916  3     AND .J NEQ .INDEX
;  500          0917  3     AND .ALLOC_TABLE_LBN[.J] + .ALLOC_TABLE_CNT[.J] GTRU .ALLOC_TABLE_LBN[.INDEX]
;  501          0918  3     AND .ALLOC_TABLE_LBN[.J] LSSU .ALLOC_TABLE_CNT[.INDEX] + .ALLOC_TABLE_LBN[.INDEX]
;  502          0919  3     THEN EXITLOOP .J;
;  503          0920  3     END
```

```
;  504         0921  3
;  505         0922  1 END;                                        ! end of routine CHECK_ALLOC


                                                                     .EXTRN  ALLOC_MAX

                                              000C 00000 CHECK_ALLOC:
                                                                     .WORD   Save R2,R3                               ; 0864
                        53       0000G  CF   9E 00002                MOVAB   ALLOC_TABLE_LBN, R3                      ;
                        50              01   CE 00007                MNEGL   #1, J                                    ; 0913
                                        2F   11 0000A                BRB     2$                                       ;
                        51       0000GCF40   D0 0000C 1$:            MOVL    ALLOC_TABLE_CNT[J], R1                   ; 0915
                                        27   13 00012                BEQL    2$                                       ;
                   04   AC              50   D1 00014                CMPL    J, INDEX                                 ; 0916
                                        21   13 00018                BEQL    2$                                       ;
          52          6340              51   C1 0001A                ADDL3   R1, ALLOC_TABLE_LBN[J], R2               ; 0917
                        51       04     AC   D0 0001F                MOVL    INDEX, R1                                ;
                      6341              52   D1 00023                CMPL    R2, ALLOC_TABLE_LBN[R1]                  ;
                                        12   1B 00027                BLEQU   2$                                       ;
                        51       04     AC   D0 00029                MOVL    INDEX, R1                                ; 0918
          51     0000GCF41            6341   C1 0002D                ADDL3   ALLOC_TABLE_LBN[R1], ALLOC_TABLE_CNT[R1], - ;
                                                                             R1                                       ;
                        51            6340   D1 00035                CMPL    ALLOC_TABLE_LBN[J], R1                   ;
                                        0B   1F 00039                BLSSU   3$                                       ;
          C9            50 000000000G   8F   F3 0003B 2$:            AOBLEQ  #ALLOC_MAX, J, 1$                        ; 0913
                        50              01   CE 00043                MNEGL   #1, R0                                   ;
                                             04 00046 3$:            RET                                              ; 0922

; Routine Size:  71 bytes,    Routine Base:  $CODE$ + 02C7


;  506         0923  1
;  507         0924  1 END
;  508         0925  0 ELUDOM



                                                                     .EXTRN  LIB$SIGNAL, LIB$STOP

;                                   PSECT SUMMARY
;
;       Name                     Bytes                        Attributes
;
;  $OWN$                             4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                          782  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                           Library Statistics
;
;                                          -------- Symbols --------      Pages        Processing
;       File                               Total    Loaded   Percent     Mapped       Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619         8         0      1000         00:02.0
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         782 code + 4 data bytes
; Run Time:        00:18.9
; Elapsed Time:    00:42.6
; Lines/CPU Min:    2933
; Lexemes/CPU-Min: 29936
; Memory Used:  132 pages
; Compilation Complete
```

0186 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INIT

HLDPRINT
LIS

IMGDMPDEF
SDL

INIT
MAP

HLDDATA
LIS

INIBAD
LIS

INIDEF
B32

HLDMAIN
LIS

ANALIMDMP
LIS

IMGDMP

ANALIMDMP
MAP

IMGDMP
MAP

HLDIO
LIS

DISPIMDMP
LIS

HLDFILE
LIS

HLDTASK
LIS

IMGDMP
LIS

INIALL
LIS