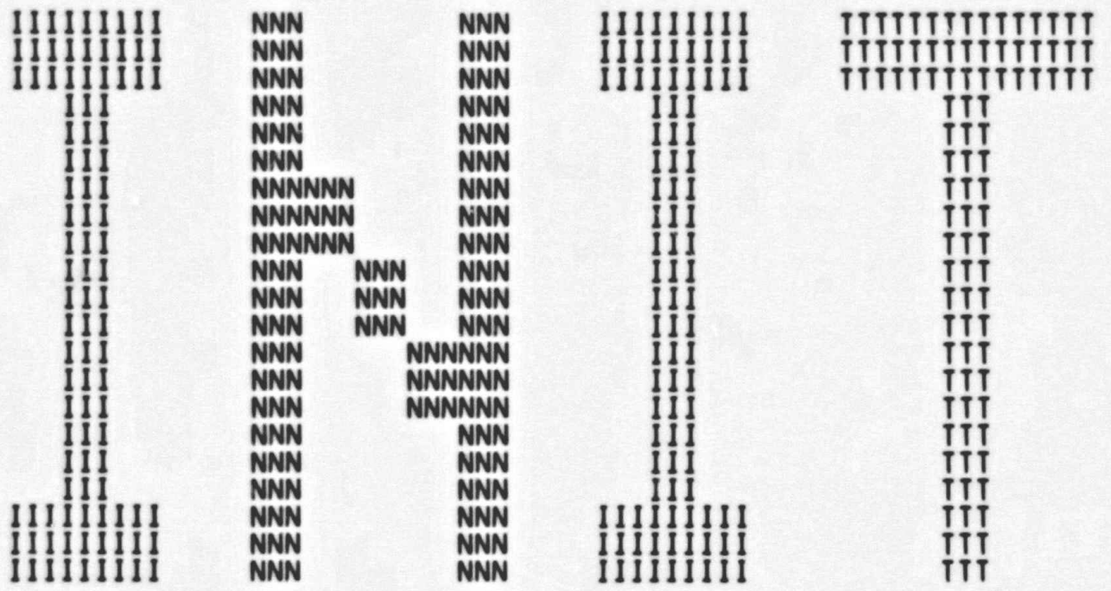
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNNNNN NNN III TTT
III NNNNNN NNN III TTT
III NNNNNN NNN III TTT
III NNN NNN NNN III TTT
III NNN NNN NNN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

```
 IIIIIII    NN       NN    IIIIIII    NN       NN  DDDDDDDD    XX       XX
 IIIIIII    NN       NN    IIIIIII    NN       NN  DDDDDDDD    XX       XX
   II       NN       NN      II       NN       NN  DD      DD  XX       XX
   II       NN       NN      II       NN       NN  DD      DD  XX       XX
   II       NNNN     NN      II       NNNN     NN  DD      DD    XX   XX
   II       NNNN     NN      II       NNNN     NN  DD      DD    XX   XX
   II       NN  NN   NN      II       NN  NN   NN  DD      DD      XX
   II       NN  NN   NN      II       NN  NN   NN  DD      DD      XX
   II       NN    NNNN      II       NN    NNNN  DD      DD    XX   XX
   II       NN    NNNN      II       NN    NNNN  DD      DD    XX   XX
   II       NN       NN      II       NN       NN  DD      DD  XX       XX     ....
   II       NN       NN      II       NN       NN  DD      DD  XX       XX     ....
 IIIIIII    NN       NN    IIIIIII    NN       NN  DDDDDDDD    XX       XX     ....
 IIIIIII    NN       NN    IIIIIII    NN       NN  DDDDDDDD    XX       XX     ....

LL             IIIIIII      SSSSSSSS
LL             IIIIIII      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     IIIIIII    SSSSSSSS
LLLLLLLLLL     IIIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE ININDX (
;     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 initialize the contents
;    38          0038  1 !       of a disk's index file: boot and home blocks, bitmap, and the
;    39          0039  1 !       initial file headers.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:
;    42          0042  1 !
;    43          0043  1 !       STARLET operating system, including privileged system services
;    44          0044  1 !       and internal exec routines.
;    45          0045  1 !
;    46          0046  1 !--
;    47          0047  1 !
;    48          0048  1 !
;    49          0049  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  14-Nov-1977  10:16
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-005 MCN0140         Maria del C. Nasr       30-Nov-1983
;    54          0054  1 !               Define LABEL_STRING and USER_NAME as BBLOCK descriptors.
;    55          0055  1 !               Default RECORD_PROT value since qualifier was never
;    56          0056  1 !               implemented.
;    57          0057  1 !
```

```
;    58          0058  1 !       V03-004 ACG0362         Andrew C. Goldstein,    27-Sep-1983  15:07
;    59          0059  1 !               Fix index file highwater mark problems
;    60          0060  1 !
;    61          0061  1 !       V03-003 ACG0332         Andrew C. Goldstein,    5-May-1983  14:37
;    62          0062  1 !               Add correct highwater mark initialization
;    63          0063  1 !
;    64          0064  1 !       V03-002 STJ3094         Steven T. Jeffreys,     27-Apr-1983
;    65          0065  1 !               Add support for /[NO]ERASE and /[NO]HIGHWATER.
;    66          0066  1 !
;    67          0067  1 !       V03-001 ACG0325         Andrew C. Goldstein,    4-Apr-1983  16:31
;    68          0068  1 !               Add high water mark field and file name extension
;    69          0069  1 !
;    70          0070  1 !       V02-004 ACG0240         Andrew C. Goldstein,    11-Dec-1981  22:17
;    71          0071  1 !               Make default file protection more restrictive
;    72          0072  1 !
;    73          0073  1 !       V02-003 ACG0185         Andrew C. Goldstein,    3-Feb-1981  21:03
;    74          0074  1 !               File structure updates; e.g., back links
;    75          0075  1 !
;    76          0076  1 !       V0102   ACG0075         Andrew C. Goldstein,    19-Oct-1979  17:51
;    77          0077  1 !               Add pack serial number to home block
;    78          0078  1 !
;    79          0079  1 !       V0101   ACG0017         Andrew C. Goldstein,  18-Jan-1979  11:49
;    80          0080  1 !               Fix generation of format 3 map pointers
;    81          0081  1 !
;    82          0082  1 !       V0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  21:27
;    83          0083  1 !               Previous revision history moved to [INIT.SRC]INIT.REV
;    84          0084  1 !**
;    85          0085  1
;    86          0086  1
;    87          0087  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    88          0088  1 REQUIRE 'SRC$:INIDEF.B32';
;    89          0379  1 REQUIRE 'LIBD$:[VMSLIB.OBJ]INITMSG.B32';
;    90          0511  1
;    91          0512  1
;    92          0513  1 FORWARD ROUTINE
;    93          0514  1         INIT_INDEX      : NOVALUE,      ! main index file initialization
;    94          0515  1         WRITE_HOMEBLOCK : NOVALUE,      ! checksum and write home block
;    95          0516  1         MAKE_POINTER    : NOVALUE;      ! construct retrieval pointer
```

```
;    97          0517  1 !+
;    98          0518  1 !
;    99          0519  1 ! Own storage.
;   100          0520  1 !
;   101          0521  1 ! Boot program. The following PDP-11 program will type out the attached
;   102          0522  1 ! message when the volume is booted on a PDP-11, informing the user that
;   103          0523  1 ! this is not a system disk.
;   104          0524  1 !
;   105          0525  1 !-
;   106          0526  1
;   107          0527  1 BIND
;   108          0528  1         BOOT_PROGRAM    = UPLIT WORD (
;   109          0529  1
;   110          0530  1 %O'000240',             !  BOOTBK: NOP                     ; NOP IDENTIFIES BOOT BLOCK
;   111          0531  1 %O'012706', %O'001000', !          MOV     #1000,SP      ; SET TEMP STACK
;   112          0532  1 %O'010700',             !          MOV     PC,R0         ; SET ADDRESS
;   113          0533  1 %O'062700', %O'000036', !          ADD     #BOTMSG-.,R0  ; OF MESSAGE
;   114          0534  1 %O'112001',             !  10$:    MOVB    (R0)+,R1      ; GET NEXT CHARACTER
;   115          0535  1 %O'001403',             !          BEQ     20$           ; END
;   116          0536  1 %O'004767', %O'000006', !          CALL    TYPIT         ; NO, PRINT IT
;   117          0537  1 %O'000773',             !          BR      10$           ; LOOP FOR NEXT CHARACTER
;   118          0538  1 %O'000005',             !  20$:    RESET                 ;
;   119          0539  1 %O'000000',             !          HALT                  ; HALT
;   120          0540  1                         !
;   121          0541  1                         !
;   122          0542  1 %O'110137', %O'177566', !  TYPIT:  MOVB    R1,@#TPB      ; PRINT CHARACTER
;   123          0543  1 %O'105737', %O'177564', !  10$:    TSTB    @#TPS         ; DONE?
;   124          0544  1 %O'100375',             !          BPL     10$           ; NO, WAIT
;   125          0545  1 %O'000207'              !          RETURN                ;
;   126          0546  1                         !
;   127          0547  1                         !
;   128          0548  1                         !  BOTMSG:
;   129          0549  1                         !      BOTMSG:
;   130          0550  1                     );
;   131          0551  1
;   132          0552  1 LITERAL
;   133          0553  1         BOOT_PROG_LEN   = 38;
;   134          0554  1
;   135          0555  1 !+
;   136          0556  1 !
;   137          0557  1 ! Boot message. Contains the volume label.
;   138          0558  1 !
;   139          0559  1 !-
;   140          0560  1
;   141          0561  1 BIND
;   142          0562  1         BOOT_MESSAGE    = UPLIT BYTE (13, 10, 10,
;   143          0563  1                                       'is not a system disk', 13, 10, 10, 0);
;   144          0564  1
;   145          0565  1 LITERAL
;   146          0566  1         BOOT_MESG_LEN   = 40;
;   147          0567  1
;   148          0568  1 MACRO
;   149          0569  1         BTB$T_VOLNAME   = 38, 0, 0, 0%; ! volume label in boot block message
;   150          0570  1
;   151          0571  1 !
;   152          0572  1 ! Volume format name string
;   153          0573  1 !
```

```
;  154          0574  1
;  155          0575  1 BIND
;  156          0576  1         FORMAT_NAME     = UPLIT BYTE ('DECFILE11B  ');
;  157          0577  1
;  158          0578  1 !+
;  159          0579  1 !
;  160          0580  1 ! Initial file header. The core image file is used since it is the first
;  161          0581  1 ! one written. Note that this must be updated whenever fields are added
;  162          0582  1 ! to the file header.
;  163          0583  1 !
;  164          0584  1 !-
;  165          0585  1
;  166          0586  1 $ASSUME (FH2$C_LENGTH, EQL, 80)
;  167          0587  1 $ASSUME (FI2$C_LENGTH, EQL, 120)
;  168          0588  1
;  169          0589  1 BIND
;  170          0590  1         INITIAL_HEADER  = UPLIT (
;  171          0591  1                                                         ! HEADER area
;  172          0592  1         BYTE (FH2$C_LENGTH / 2),                        ! ident area offset
;  173          0593  1         BYTE ((FH2$C_LENGTH + FI2$C_LENGTH)/2),         ! map area offset
;  174          0594  1         BYTE ($BYTEOFFSET (FH2$W_CHECKSUM)/2),          ! access control list offset
;  175          0595  1         BYTE ($BYTEOFFSET (FH2$W_CHECKSUM)/2),          ! reserved area offset
;  176          0596  1         WORD (0),                                       ! file segment number
;  177          0597  1         BYTE (1, 2),                                    ! structure version and level
;  178          0598  1         WORD (5, 5, 0),                                 ! file ID
;  179          0599  1         WORD (0, 0, 0),                                 ! extension file ID
;  180          0600  1         BYTE (FAT$C_FIXED),                             ! fixed length record type
;  181          0601  1         BYTE (0),                                       ! no record attributes
;  182          0602  1         WORD (512),                                     ! record size
;  183          0603  1         LONG (0, 1^16),                                 ! HIBLK and EFBLK
;  184          0604  1         WORD (0),                                       ! EOF byte offset
;  185          0605  1         BYTE (0, 0),                                    ! bucket size & VFC length
;  186          0606  1         WORD (512),                                     ! maximum record length
;  187          0607  1         WORD (0),                                       ! default extend size
;  188          0608  1         WORD (0, 0, 0, 0, 0, 0),                        ! unused record attributes
;  189          0609  1         LONG (0),                                       ! file characteristics
;  190          0610  1         WORD (0),                                       ! record protection
;  191          0611  1         BYTE (0, 0),                                    ! mapwords in use & access mode
;  192          0612  1         LONG (0),                                       ! file owner UIC
;  193          0613  1         WORD (0),                                       ! file protection
;  194          0614  1         WORD (4, 4, 0),                                 ! directory back link
;  195          0615  1         WORD (0,0),                                     ! journal flags and spare
;  196          0616  1         LONG (1),                                       ! high water mark
;  197          0617  1
;  198          0618  1                                                         ! IDENT area
;  199          0619  1         BYTE ('CORIMG.SYS;1        '),                  ! file name, type and version
;  200          0620  1         WORD (1),                                       ! revision number
;  201          0621  1         LONG (0, 0, 0, 0, 0, 0, 0, 0),                  ! dates
;  202          0622  1         REP FI2$S_FILENAMEXT OF BYTE (' ')              ! file name extension
;  203          0623  1
;  204          0624  1         );
```

```
;  206          0625  1 GLOBAL ROUTINE INIT_INDEX : NOVALUE =
;  207          0626  1
;  208          0627  1 !++
;  209          0628  1 !
;  210          0629  1 ! FUNCTIONAL DESCRIPTION:
;  211          0630  1 !
;  212          0631  1 !       This routine initializes the contents of the disk's index file.
;  213          0632  1 !       It writes a dummy boot block, the home blocks, index file bitmap,
;  214          0633  1 !       and the initial headers.
;  215          0634  1 !
;  216          0635  1 !
;  217          0636  1 ! CALLING SEQUENCE:
;  218          0637  1 !       INIT_INDEX ()
;  219          0638  1 !
;  220          0639  1 ! INPUT PARAMETERS:
;  221          0640  1 !       NONE
;  222          0641  1 !
;  223          0642  1 ! IMPLICIT INPUTS:
;  224          0643  1 !       parser data base
;  225          0644  1 !       allocation table in INIDSK
;  226          0645  1 !
;  227          0646  1 ! OUTPUT PARAMETERS:
;  228          0647  1 !       NONE
;  229          0648  1 !
;  230          0649  1 ! IMPLICIT OUTPUTS:
;  231          0650  1 !       NONE
;  232          0651  1 !
;  233          0652  1 ! ROUTINE VALUE:
;  234          0653  1 !       NONE
;  235          0654  1 !
;  236          0655  1 ! SIDE EFFECTS:
;  237          0656  1 !       index file blocks written
;  238          0657  1 !
;  239          0658  1 !--
;  240          0659  1
;  241          0660  2 BEGIN
;  242          0661  2
;  243          0662  2 BUILTIN
;  244          0663  2         ROT;
;  245          0664  2
;  246          0665  2 LOCAL
;  247          0666  2         DATE_TIME       : VECTOR [2],   ! buffer for current date/time
;  248          0667  2         LBN,                            ! current LBN
;  249          0668  2         MAP_COUNT,                      ! count field of map pointer
;  250          0669  2         MAP_LBN;                        ! start LBN of current map pointer
;  251          0670  2
;  252          0671  2 EXTERNAL
;  253          0672  2         INIT_OPTIONS    : BITVECTOR,    ! command options
;  254          0673  2         BUFFER          : BBLOCK,       ! I/O buffer
;  255          0674  2         VOLUME_SIZE,                    ! size of volume rounded to next cluster
;  256          0675  2         PROTECTION,                     ! volume protection
;  257          0676  2         FILE_PROT,                      ! default file protection
;  258          0677  2         MAXIMUM,                        ! maximum number of files on volume
;  259          0678  2         CLUSTER,                        ! volume cluster factor
;  260          0679  2         OWNER_UIC,                      ! volume owner
;  261          0680  2         EXTENSION,                      ! default file extend
;  262          0681  2         WINDOW,                         ! default window size
```

```
;  263          0682  2          ACCESSED,                          ! default directory LRU limit
;  264          0683  2          SERIAL_NUMBER,                     ! pack serial number
;  265          0684  2          BADBLOCK_TOTAL,                    ! count of bad blocks on volume
;  266          0685  2          ALLOC_TABLE_CNT : VECTOR,          ! allocation count table
;  267          0686  2          ALLOC_TABLE_LBN : VECTOR,          ! allocation LBN table
;  268          0687  2          BADBLOCK_CNT    : VECTOR,          ! bad block count table
;  269          0688  2          BADBLOCK_LBN    : VECTOR,          ! bad LBN table
;  270          0689  2          BOOTBLOCK_CNT,                     ! boot block cluster block count
;  271          0690  2          BOOTBLOCK_LBN,                     ! boot block cluster LBN
;  272          0691  2          HOMEBLOCK1_CNT,                    ! home block 1 cluster block count
;  273          0692  2          HOMEBLOCK1_LBN,                    ! home block 1 cluster LBN
;  274          0693  2          HOMEBLOCK2_CNT,                    ! home block 2 cluster block count
;  275          0694  2          HOMEBLOCK2_LBN,                    ! home block 2 cluster LBN
;  276          0695  2          IDXHDR2_CNT,                       ! secondary index file header count
;  277          0696  2          IDXHDR2_LBN,                       ! secondary index file header LBN
;  278          0697  2          IDXFILE_CNT,                       ! initial index file count
;  279          0698  2          IDXFILE_LBN,                       ! initial index file LBN
;  280          0699  2          BITMAP_CNT,                        ! storage bitmap block count
;  281          0700  2          BITMAP_LBN,                        ! storage bitmap LBN
;  282          0701  2          MFD_CNT,                           ! MFD block count
;  283          0702  2          MFD_LBN,                           ! MFD LBN
;  284          0703  2          REAL_HOMEBLOCK,                    ! LBN of secondary home block
;  285          0704  2          LABEL_STRING    : BBLOCK [DSC$C_S_BLN], ! string descriptor of volume label
;  286          0705  2          USER_NAME       : BBLOCK [DSC$C_S_BLN]; ! string descriptor of user name
;  287          0706  2
;  288          0707  2      EXTERNAL LITERAL
;  289          0708  2          BOOTBLOCK_IDX   : UNSIGNED (6), ! allocation table boot block index
;  290          0709  2          IDXFILE_IDX     : UNSIGNED (6); ! allocation table index file index
;  291          0710  2
;  292          0711  2      BIND
;  293          0712  2          DEF_REC_PROT    = UPLIT ( %X'FE00' ),          ! default record prot
;  294          0713  2          IDENT_AREA      = BUFFER + FH2$C_LENGTH : BBLOCK;
;  295          0714  2
;  296          0715  2      EXTERNAL ROUTINE
;  297          0716  2          CHECKSUM2,                         ! compute block checksum
;  298          0717  2          WRITE_BLOCK;                       ! write block to disk
;  299          0718  2
;  300          0719  2
;  301          0720  2      ! First block to write is the boot block. Set up the message routine for
;  302          0721  2      ! the -11 and build the message.
;  303          0722  2      !
;  304          0723  2
;  305          0724  2      CH$COPY (BOOT_PROG_LEN, BOOT_PROGRAM,
;  306          0725  2               BOOT_MESG_LEN, BOOT_MESSAGE,
;  307          0726  2               0, 512, BUFFER);
;  308          0727  2      CH$MOVE ( .LABEL_STRING [DSC$W_LENGTH],
;  309          0728  2                .LABEL_STRING [DSC$A_POINTER],
;  310          0729  2                BUFFER[BTB$T_VOLNAME] );
;  311          0730  2
;  312          0731  2      WRITE_BLOCK (.BOOTBLOCK_LBN, BUFFER);
;  313          0732  2
;  314          0733  2      ! Now construct the home block. It gets written to the remainder of the boot
;  315          0734  2      ! block cluster and to the two home block clusters.
;  316          0735  2      !
;  317          0736  2
;  318          0737  2      $GETTIM (TIMADR = DATE_TIME[0]);
;  319          0738  2      CH$FILL (0, 512, BUFFER);
```

```
;  320          0739  2
;  321          0740  2 BUFFER[HM2$L_HOMELBN]   = .BOOTBLOCK_LBN + 1;
;  322          0741  2 BUFFER[HM2$L_ALHOMELBN] = .REAL_HOMEBLOCK;
;  323          0742  2 BUFFER[HM2$L_ALTIDXLBN] = .IDXHDR2_LBN;
;  324          0743  2 BUFFER[HM2$B_STRUCVER]  = 1;
;  325          0744  2 BUFFER[HM2$B_STRUCLEV]  = 2;
;  326          0745  2 BUFFER[HM2$W_CLUSTER]   = .CLUSTER;
;  327          0746  2 BUFFER[HM2$W_HOMEVBN]   = 2;
;  328          0747  2 BUFFER[HM2$W_ALHOMEVBN] = .REAL_HOMEBLOCK - .HOMEBLOCK2_LBN + .CLUSTER * 2 + 1;
;  329          0748  2 BUFFER[HM2$W_ALTIDXVBN] = .CLUSTER * 3 + 1;
;  330          0749  2 BUFFER[HM2$W_IBMAPVBN]  = .CLUSTER * 4 + 1;
;  331          0750  2 BUFFER[HM2$L_IBMAPLBN]  = .IDXFILE_LBN;
;  332          0751  2 BUFFER[HM2$L_MAXFILES]  = .MAXIMUM;
;  333          0752  2 BUFFER[HM2$W_IBMAPSIZE] = (.MAXIMUM + 4095) / 4096;
;  334          0753  2 BUFFER[HM2$W_RESFILES]  = 9;
;  335          0754  2 BUFFER[HM2$L_VOLOWNER]  = .OWNER_UIC;
;  336          0755  2 BUFFER[HM2$W_PROTECT]   = .PROTECTION;
;  337          0756  2 IF .INIT_OPTIONS[OPT_READCHECK]
;  338          0757  2 THEN BUFFER[HM2$V_READCHECK] = 1;
;  339          0758  2 IF .INIT_OPTIONS[OPT_WRITECHECK]
;  340          0759  2 THEN BUFFER[HM2$V_WRITCHECK] = 1;
;  341          0760  2 BUFFER[HM2$W_FILEPROT]  = .FILE_PROT;
;  342          0761  2 BUFFER[HM2$W_RECPROT]   = .DEF_REC_PROT;
;  343          0762  2 (BUFFER[HM2$Q_CREDATE])<0,32>   = .DATE_TIME[0];
;  344          0763  2 (BUFFER[HM2$Q_CREDATE]+4)<0,32> = .DATE_TIME[1];
;  345          0764  2 BUFFER[HM2$B_WINDOW]    = .WINDOW;
;  346          0765  2 BUFFER[HM2$B_LRU_LIM]   = .ACCESSED;
;  347          0766  2 BUFFER[HM2$W_EXTEND]    = .EXTENSION;
;  348          0767  2 BUFFER[HM2$L_SERIALNUM] = .SERIAL_NUMBER;
;  349          0768  2 IF .INIT_OPTIONS[OPT_ERASE]
;  350          0769  2 THEN BUFFER[HM2$V_ERASE] = 1;
;  351          0770  2 IF .INIT_OPTIONS[OPT_NOHIGHWATER]
;  352          0771  2 THEN BUFFER[HM2$V_NOHIGHWATER] = 1;
;  353          0772  2
;  354          0773  2 CH$FILL (32, HM2$S_STRUCNAME, BUFFER[HM2$T_STRUCNAME]);
;  355          0774  2 CH$COPY (.LABEL_STRING [DSC$W_LENGTH], .LABEL_STRING [DSC$A_POINTER],
;  356          0775  2          32, HM2$S_VOLNAME, BUFFER[HM2$T_VOLNAME]);
;  357          0776  2 CH$COPY (.USER_NAME [DSC$W_LENGTH], .USER_NAME [DSC$A_POINTER],
;  358          0777  2          32, HM2$S_OWNERNAME, BUFFER[HM2$T_OWNERNAME]);
;  359          0778  2 CH$MOVE (HM2$S_FORMAT, FORMAT_NAME, BUFFER[HM2$T_FORMAT]);
;  360          0779  2
;  361          0780  2 DECR J FROM .CLUSTER-1 TO 1 DO
;  362          0781  2     WRITE_HOMEBLOCK ();
;  363          0782  2
;  364          0783  2 BUFFER[HM2$L_HOMELBN] = .HOMEBLOCK1_LBN;
;  365          0784  2 DECR J FROM .CLUSTER TO 1 DO
;  366          0785  2     WRITE_HOMEBLOCK ();
;  367          0786  2
;  368          0787  2 BUFFER[HM2$L_HOMELBN] = .HOMEBLOCK2_LBN;
;  369          0788  2 DECR J FROM .CLUSTER TO 1 DO
;  370          0789  2     WRITE_HOMEBLOCK ();
;  371          0790  2
;  372          0791  2 ! Now write out the initial index file bitmap. The first block contains the
;  373          0792  2 ! reserved files marked in use; the rest are all zero.
;  374          0793  2 !
;  375          0794  2
;  376          0795  2 CH$FILL (0, 512, BUFFER);
```

```
;  377          0796  2 BUFFER<0,32> = %B'111111111';
;  378          0797  2 LBN = .IDXFILE_LBN;
;  379          0798  2 WRITE_BLOCK (.LBN, BUFFER);
;  380          0799  2
;  381          0800  2 BUFFER<0,32> = 0;
;  382          0801  2 DECR J FROM (.MAXIMUM+4095)/4096-1 TO 1 DO
;  383          0802  3     BEGIN
;  384          0803  3     LBN = .LBN + 1;
;  385          0804  3     WRITE_BLOCK (.LBN, BUFFER);
;  386          0805  2     END;
;  387          0806  2
;  388          0807  2 ! Construct and write the initial core image file header.
;  389          0808  2 !
;  390          0809  2
;  391          0810  2 CH$COPY (FH2$C_LENGTH+FI2$C_LENGTH, INITIAL_HEADER,
;  392          0811  2             0, 512, BUFFER);
;  393          0812  2 BUFFER[FH2$L_FILEOWNER] = .OWNER_UIC;
;  394          0813  2 BUFFER[FH2$W_FILEPROT] = .FILE_PROT;
;  395          0814  2 BUFFER[FH2$W_RECPROT] = .DEF_REC_PROT;
;  396          0815  2 (IDENT_AREA[FI2$Q_CREDATE]) = .DATE_TIME[0];
;  397          0816  2 (IDENT_AREA[FI2$Q_CREDATE]+4) = .DATE_TIME[1];
;  398          0817  2 (IDENT_AREA[FI2$Q_REVDATE]) = .DATE_TIME[0];
;  399          0818  2 (IDENT_AREA[FI2$Q_REVDATE]+4) = .DATE_TIME[1];
;  400          0819  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  401          0820  2 WRITE_BLOCK (.LBN + 5, BUFFER);
;  402          0821  2
;  403          0822  2 ! Turn the header into the continuation file header and write it.
;  404          0823  2 !
;  405          0824  2
;  406          0825  2 BUFFER[FH2$W_FID_NUM] = 7;
;  407          0826  2 BUFFER[FH2$W_FID_SEQ] = 7;
;  408          0827  2 CH$MOVE (6, UPLIT BYTE ('CONTIN'), IDENT_AREA[FI2$T_FILENAME]);
;  409          0828  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  410          0829  2 WRITE_BLOCK (.LBN + 7, BUFFER);
;  411          0830  2
;  412          0831  2 ! Turn the header into the volume set list file header and write it.
;  413          0832  2 !
;  414          0833  2
;  415          0834  2 BUFFER[FH2$W_FID_NUM] = 6;
;  416          0835  2 BUFFER[FH2$W_FID_SEQ] = 6;
;  417          0836  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_RSIZE] = 64;
;  418          0837  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_MAXREC] = 64;
;  419          0838  2 CH$MOVE (6, UPLIT BYTE ('VOLSET'), IDENT_AREA[FI2$T_FILENAME]);
;  420          0839  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  421          0840  2 WRITE_BLOCK (.LBN + 6, BUFFER);
;  422          0841  2
;  423          0842  2 ! Turn the header into the backup journal file header and write it.
;  424          0843  2 !
;  425          0844  2
;  426          0845  2 BUFFER[FH2$W_FID_NUM] = 8;
;  427          0846  2 BUFFER[FH2$W_FID_SEQ] = 8;
;  428          0847  2 CH$MOVE (6, UPLIT BYTE ('BACKUP'), IDENT_AREA[FI2$T_FILENAME]);
;  429          0848  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  430          0849  2 WRITE_BLOCK (.LBN + 8, BUFFER);
;  431          0850  2
;  432          0851  2 ! Turn the header into the pending bad block log file header and write it.
;  433          0852  2 !
```

```
;  434          0853  2
;  435          0854  2 BUFFER[FH2$W_FID_NUM] = 9;
;  436          0855  2 BUFFER[FH2$W_FID_SEQ] = 9;
;  437          0856  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_RSIZE] = 16;
;  438          0857  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_MAXREC] = 16;
;  439          0858  2 CH$MOVE (6, UPLIT BYTE ('BADLOG'), IDENT_AREA[FI2$T_FILENAME]);
;  440          0859  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  441          0860  2 WRITE_BLOCK (.LBN + 9, BUFFER);
;  442          0861  2
;  443          0862  2 ! Turn the header into the index file header and write it.
;  444          0863  2 !
;  445          0864  2
;  446          0865  2 BUFFER[FH2$W_FID_NUM] = 1;
;  447          0866  2 BUFFER[FH2$W_FID_SEQ] = 1;
;  448          0867  2 BUFFER[FH2$L_HIGHWATER] = .CLUSTER*4 + .IDXFILE_CNT + 1;
;  449          0868  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_RSIZE] = 512;
;  450          0869  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_MAXREC] = 512;
;  451          0870  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT (.CLUSTER*4 + .IDXFILE_CNT, 16);
;  452          0871  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT (.CLUSTER*4 + (.MAXIMUM+4095)/4096 + 9 + 1, 16);
;  453          0872  2 CH$MOVE (6, UPLIT BYTE ('INDEXF'), IDENT_AREA[FI2$T_FILENAME]);
;  454          0873  2 MAP_COUNT = .BOOTBLOCK_CNT;
;  455          0874  2 MAP_LBN = .BOOTBLOCK_LBN;
;  456          0875  2 INCR J FROM BOOTBLOCK_IDX + 1 TO IDXFILE_IDX DO
;  457          0876  3 BEGIN
;  458          0877  3     IF .MAP_COUNT + .MAP_LBN EQL .ALLOC_TABLE_LBN[.J]
;  459          0878  3     THEN
;  460          0879  3         MAP_COUNT = .MAP_COUNT + .ALLOC_TABLE_CNT[.J]
;  461          0880  3     ELSE
;  462          0881  4         BEGIN
;  463          0882  4         MAKE_POINTER (.MAP_COUNT, .MAP_LBN);
;  464          0883  4         MAP_COUNT = .ALLOC_TABLE_CNT[.J];
;  465          0884  4         MAP_LBN = .ALLOC_TABLE_LBN[.J];
;  466          0885  3         END;
;  467          0886  2     END;
;  468          0887  2 MAKE_POINTER (.MAP_COUNT, .MAP_LBN);
;  469          0888  2
;  470          0889  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  471          0890  2 WRITE_BLOCK (.LBN + 1, BUFFER);
;  472          0891  2 DECR J FROM .CLUSTER-1 TO 0
;  473          0892  2 DO WRITE_BLOCK (.IDXHDR2_LBN+.J, BUFFER);
;  474          0893  2
;  475          0894  2 ! Turn the file header into the bad block file header and write it.
;  476          0895  2 !
;  477          0896  2
;  478          0897  2 CH$FILL (0, 512-FH2$C_LENGTH-FI2$C_LENGTH, BUFFER+FH2$C_LENGTH+FI2$C_LENGTH);
;  479          0898  2 BUFFER[FH2$B_MAP_INUSE] = 0;
;  480          0899  2 BUFFER[FH2$W_FID_NUM] = 3;
;  481          0900  2 BUFFER[FH2$W_FID_SEQ] = 3;
;  482          0901  2
;  483          0902  2 MAP_COUNT = 0;
;  484          0903  2 INCR J FROM 0 TO .BADBLOCK_TOTAL-1 DO
;  485          0904  2     MAP_COUNT = .MAP_COUNT + .BADBLOCK_CNT[.J];
;  486          0905  2 BUFFER[FH2$L_HIGHWATER] = .MAP_COUNT + 1;
;  487          0906  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT (.MAP_COUNT, 16);
;  488          0907  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT (.MAP_COUNT+1, 16);
;  489          0908  2
;  490          0909  2 CH$MOVE (6, UPLIT BYTE ('BADBLK'), IDENT_AREA[FI2$T_FILENAME]);
```

```
;   491          0910  2 INCR J FROM 0 TO .BADBLOCK_TOTAL-1 DO
;   492          0911  3     BEGIN
;   493          0912  3     IF .BUFFER[FH2$B_MAP_INUSE] GTR (512 - FH2$C_LENGTH - FI2$C_LENGTH - 2) / 2 - 4
;   494          0913  3     THEN ERR_EXIT (INIT$_MAXBAD);
;   495          0914  3     MAKE_POINTER (.BADBLOCK_CNT[.J], .BADBLOCK_LBN[.J]);
;   496          0915  2     END;
;   497          0916  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;   498          0917  2 WRITE_BLOCK (.LBN + 3, BUFFER);
;   499          0918  2
;   500          0919  2 ! Turn the file header into the storage map file header and write it.
;   501          0920  2 !
;   502          0921  2
;   503          0922  2 CH$FILL (0, 512-FH2$C_LENGTH-FI2$C_LENGTH, BUFFER+FH2$C_LENGTH+FI2$C_LENGTH);
;   504          0923  2 BUFFER[FH2$B_MAP_INUSE] = 0;
;   505          0924  2 BUFFER[FH2$W_FID_NUM] = 2;
;   506          0925  2 BUFFER[FH2$W_FID_SEQ] = 2;
;   507          0926  2 BUFFER[FH2$V_CONTIG] = 1;
;   508          0927  2 BUFFER[FH2$L_HIGHWATER] = (.VOLUME_SIZE/.CLUSTER+4095)/4096 + 2;
;   509          0928  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT (.BITMAP_CNT, 16);
;   510          0929  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT ((.VOLUME_SIZE/.CLUSTER+4095)/4096 + 2, 16);
;   511          0930  2
;   512          0931  2 CH$MOVE (6, UPLIT BYTE ('BITMAP'), IDENT_AREA[FI2$T_FILENAME]);
;   513          0932  2 MAKE_POINTER (.BITMAP_CNT, .BITMAP_LBN);
;   514          0933  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;   515          0934  2 WRITE_BLOCK (.LBN + 2, BUFFER);
;   516          0935  2
;   517          0936  2 ! Turn the file header into the MFD header and write it.
;   518          0937  2 !
;   519          0938  2
;   520          0939  2 CH$FILL (0, 512-FH2$C_LENGTH-FI2$C_LENGTH, BUFFER+FH2$C_LENGTH+FI2$C_LENGTH);
;   521          0940  2 BUFFER[FH2$B_MAP_INUSE] = 0;
;   522          0941  2 BUFFER[FH2$W_FID_NUM] = 4;
;   523          0942  2 BUFFER[FH2$W_FID_SEQ] = 4;
;   524          0943  2 BUFFER[FH2$V_DIRECTORY] = 1;
;   525          0944  2 BUFFER[FH2$W_FILEPROT] = .BUFFER[FH2$W_FILEPROT] AND NOT %X'4444';
;   526          0945  2 BUFFER[FH2$L_HIGHWATER] =  2;
;   527          0946  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT (2, 16);
;   528          0947  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT (.MFD_CNT, 16);
;   529          0948  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$B_RTYPE] = FAT$C_VARIABLE;
;   530          0949  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$B_RATTRIB] = FAT$M_NOSPAN;
;   531          0950  2
;   532          0951  2 CH$MOVE (10, UPLIT BYTE ('000000.DIR'), IDENT_AREA[FI2$T_FILENAME]);
;   533          0952  2 MAKE_POINTER (.MFD_CNT, .MFD_LBN);
;   534          0953  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;   535          0954  2 WRITE_BLOCK (.LBN + 4, BUFFER);
;   536          0955  2
;   537          0956  1 END;                                    ! end of routine INIT_INDEX

                                                                .TITLE  ININDX
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

0006  09F7  0303  9401  001E  65C0  11C0  0200  15C6  00A0  00000 P.AAA:  .WORD   160, 5574, 512, 4544, 26048, 30, -27647, -  ;
      0087  80FD  FF74  8BDF  FF76  905F  0000  0005  01FB  00014                 771, 2551, 6, 507, 5, 0, -28577, -138, -  ;
                                                                          -29729, -140, -32515, 135                 ;
```

```
                                                    0A  0A  0D  00026 P.AAB:   .BYTE   13, 10, 10                                  ;
73  69  20  20  20  20  20  20  20  20  20  20  20  20  20  20  00029         .ASCII  \                 is not a system disk\         ;
64  20  6D  65  74  73  79  73  20  61  20  74  6F  6E  20  00038                                                                   ;
                                                    6B  73  69  00047                                                             ;
                                                00  0A  0A  0D  0004A         .BYTE   13, 10, 10, 0                                 ;
            20  20  42  31  31  45  4C  49  46  43  45  44  00004E P.AAC:   .ASCII  \DECFILE11B  \                                 ;
                                                                00005A         .BLKB   2                                             ;
                                                        28  00005C P.AAD:   .BYTE   40                                            ;
                                                        64  00005D         .BYTE   100                                           ;
                                                        FF  00005E         .BYTE   -1                                            ;
                                                        FF  00005F         .BYTE   -1                                            ;
                                                    0000  00060         .WORD   0                                             ;
                                                  02  01  00062         .BYTE   1, 2                                          ;
                                            0000  0005  0005  00064         .WORD   5, 5, 0                                       ;
                                            0000  0000  0000  0006A         .WORD   0, 0, 0                                       ;
                                                        01  00070         .BYTE   1                                             ;
                                                        00  00071         .BYTE   0                                             ;
                                                    0200  00072         .WORD   512                                           ;
                                        00010000  00000000  00074         .LONG   0, 65536                                      ;
                                                    0000  0007C         .WORD   0                                             ;
                                                  00  00  0007E         .BYTE   0, 0                                          ;
                                                    0200  00080         .WORD   512                                           ;
                                                    0000  00082         .WORD   0                                             ;
                                0000  0000  0000  0000  0000  0000  00084         .WORD   0, 0, 0, 0, 0, 0                              ;
                                                00000000  00090         .LONG   0                                             ;
                                                    0000  00094         .WORD   0                                             ;
                                                  00  00  00096         .BYTE   0, 0                                          ;
                                                00000000  00098         .LONG   0                                             ;
                                                    0000  0009C         .WORD   0                                             ;
                                            0000  0004  0004  0009E         .WORD   4, 4, 0                                       ;
                                                    0000  0000  000A4         .WORD   0, 0                                          ;
                                                00000001  000A8         .LONG   1                                             ;
20  20  20  31  3B  53  59  53  2E  47  4D  49  52  4F  43  000AC         .ASCII  \CORIMG.SYS;1         \                       ;
                                                20  20  20  20  20  000BB                                                         ;
                                                        0001  000C0         .WORD   1                                             ;
00000000  00000000  00000000  00000000  00000000  00000000  000C2         .LONG   0, 0, 0, 0, 0, 0, 0, 0                        ;
                                                00000000  00000000  000DA                                                         ;
                                                        20  000E2         .ASCII  \ \                                           ;
                                                        20  000E3         .ASCII  \ \                                           ;
                                                        20  000E4         .ASCII  \ \                                           ;
                                                        20  000E5         .ASCII  \ \                                           ;
                                                        20  000E6         .ASCII  \ \                                           ;
                                                        20  000E7         .ASCII  \ \                                           ;
                                                        20  000E8         .ASCII  \ \                                           ;
                                                        20  000E9         .ASCII  \ \                                           ;
                                                        20  000EA         .ASCII  \ \                                           ;
                                                        20  000EB         .ASCII  \ \                                           ;
                                                        20  000EC         .ASCII  \ \                                           ;
                                                        20  000ED         .ASCII  \ \                                           ;
                                                        20  000EE         .ASCII  \ \                                           ;
                                                        20  000EF         .ASCII  \ \                                           ;
                                                        20  000F0         .ASCII  \ \                                           ;
                                                        20  000F1         .ASCII  \ \                                           ;
                                                        20  000F2         .ASCII  \ \                                           ;
                                                        20  000F3         .ASCII  \ \                                           ;
                                                        20  000F4         .ASCII  \ \                                           ;
                                                        20  000F5         .ASCII  \ \                                           ;
```

```
                              20  000F6          .ASCII   \ \                       ;
                              20  000F7          .ASCII   \ \                       ;
                              20  000F8          .ASCII   \ \                       ;
                              20  000F9          .ASCII   \ \                       ;
                              20  000FA          .ASCII   \ \                       ;
                              20  000FB          .ASCII   \ \                       ;
                              20  000FC          .ASCII   \ \                       ;
                              20  000FD          .ASCII   \ \                       ;
                              20  000FE          .ASCII   \ \                       ;
                              20  000FF          .ASCII   \ \                       ;
                              20  00100          .ASCII   \ \                       ;
                              20  00101          .ASCII   \ \                       ;
                              20  00102          .ASCII   \ \                       ;
                              20  00103          .ASCII   \ \                       ;
                              20  00104          .ASCII   \ \                       ;
                              20  00105          .ASCII   \ \                       ;
                              20  00106          .ASCII   \ \                       ;
                              20  00107          .ASCII   \ \                       ;
                              20  00108          .ASCII   \ \                       ;
                              20  00109          .ASCII   \ \                       ;
                              20  0010A          .ASCII   \ \                       ;
                              20  0010B          .ASCII   \ \                       ;
                              20  0010C          .ASCII   \ \                       ;
                              20  0010D          .ASCII   \ \                       ;
                              20  0010E          .ASCII   \ \                       ;
                              20  0010F          .ASCII   \ \                       ;
                              20  00110          .ASCII   \ \                       ;
                              20  00111          .ASCII   \ \                       ;
                              20  00112          .ASCII   \ \                       ;
                              20  00113          .ASCII   \ \                       ;
                              20  00114          .ASCII   \ \                       ;
                              20  00115          .ASCII   \ \                       ;
                              20  00116          .ASCII   \ \                       ;
                              20  00117          .ASCII   \ \                       ;
                              20  00118          .ASCII   \ \                       ;
                              20  00119          .ASCII   \ \                       ;
                              20  0011A          .ASCII   \ \                       ;
                              20  0011B          .ASCII   \ \                       ;
                              20  0011C          .ASCII   \ \                       ;
                              20  0011D          .ASCII   \ \                       ;
                              20  0011E          .ASCII   \ \                       ;
                              20  0011F          .ASCII   \ \                       ;
                              20  00120          .ASCII   \ \                       ;
                              20  00121          .ASCII   \ \                       ;
                              20  00122          .ASCII   \ \                       ;
                              20  00123          .ASCII   \ \                       ;
                        0000FE00  00124 P.AAE:   .LONG    65024                     ;
             4E  49  54  4E  4F  43  00128 P.AAF:   .ASCII   \CONTIN\                  ;
             54  45  53  4C  4F  56  0012E P.AAG:   .ASCII   \VOLSET\                  ;
             50  55  4B  43  41  42  00134 P.AAH:   .ASCII   \BACKUP\                  ;
             47  4F  4C  44  41  42  0013A P.AAI:   .ASCII   \BADLOG\                  ;
             46  58  45  44  4E  49  00140 P.AAJ:   .ASCII   \INDEXF\                  ;
             4B  4C  42  44  41  42  00146 P.AAK:   .ASCII   \BADBLK\                  ;
             50  41  4D  54  49  42  0014C P.AAL:   .ASCII   \BITMAP\                  ;
 52  49  44  2E  30  30  30  30  30  30  00152 P.AAM:   .ASCII   \000000.DIR\              ;

                                        BOOT_PROGRAM=        P.AAA
```

```
                                                                BOOT_MESSAGE=       P.AAB
                                                                FORMAT_NAME=        P.AAC
                                                                INITIAL_HEADER=     P.AAD
                                                                DEF_REC_PROT=       P.AAE
                                                                        .EXTRN  INIT_OPTIONS, BUFFER
                                                                        .EXTRN  VOLUME_SIZE, PROTECTION
                                                                        .EXTRN  FILE_PROT, MAXIMUM
                                                                        .EXTRN  CLUSTER, OWNER_UIC
                                                                        .EXTRN  EXTENSION, WINDOW
                                                                        .EXTRN  ACCESSED, SERIAL_NUMBER
                                                                        .EXTRN  BADBLOCK_TOTAL, ALLOC_TABLE_CNT
                                                                        .EXTRN  ALLOC_TABLE_LBN
                                                                        .EXTRN  BADBLOCK_CNT, BADBLOCK_LBN
                                                                        .EXTRN  BOOTBLOCK_CNT, BOOTBLOCK_LBN
                                                                        .EXTRN  HOMEBLOCK1_CNT, HOMEBLOCK1_LBN
                                                                        .EXTRN  HOMEBLOCK2_CNT, HOMEBLOCK2_LBN
                                                                        .EXTRN  IDXHDR2_CNT, IDXHDR2_LBN
                                                                        .EXTRN  IDXFILE_CNT, IDXFILE_LBN
                                                                        .EXTRN  BITMAP_CNT, BITMAP_LBN
                                                                        .EXTRN  MFD_CNT, MFD_LBN
                                                                        .EXTRN  REAL_HOMEBLOCK, LABEL_STRING
                                                                        .EXTRN  USER_NAME, BOOTBLOCK_IDX
                                                                        .EXTRN  IDXFILE_IDX, CHECKSUM2
                                                                        .EXTRN  WRITE_BLOCK, SYS$GETTIM

                                                                        .PSECT  $CODE$,NOWRT,2

                                                        0FFC 00000      .ENTRY  INIT_INDEX, Save R2,R3,R4,R5,R6,R7,R8,R9,-      ; 0625
                                                                                R10,R11
                                                                                                                                ;
                                                    5B      0000G   CF  9E 00002        MOVAB   WRITE_BLOCK, R11            ;
                                                    5A      0000'   CF  9E 00007        MOVAB   DEF_REC_PROT, R10           ;
                                                    59      0000G   CF  9E 0000C        MOVAB   BUFFER, R9                  ;
                                                    5E              08  C2 00011        SUBL2   #8, SP                      ;
                                                69          FEDC    CA      26  28 00014    MOVC3   #38, BOOT_PROGRAM, BUFFER       ; 0724
01DA    8F                  00      FF02    CA      28  2C 0001A    MOVC5   #40, BOOT_MESSAGE, #0, #474, (R3)               ;
                                                                    63      00023                                               ;
                                26  A9      0000G   DF      0000G   CF  28 00024    MOVC3   LABEL_STRING, @LABEL_STRING+4, BUFFER+38 ; 0729
                                                                    59      DD 0002D    PUSHL   R9                              ; 0731
                                                                    0000G   CF  DD 0002F    PUSHL   BOOTBLOCK_LBN               ;
                                                                    6B      02  FB 00033    CALLS   #2, WRITE_BLOCK             ;
                                                                    5E      DD 00036    PUSHL   SP                              ; 0737
                                            00000000G   00          01  FB 00038    CALLS   #1, SYS$GETTIM                  ;
0200    8F                  00              6E          00  2C 0003F    MOVC5   #0, (SP), #0, #512, BUFFER              ; 0738
                                                                    69      00046                                               ;
                                69          0000G   CF      01  C1 00047    ADDL3   #1, BOOTBLOCK_LBN, BUFFER               ; 0740
                                              04    A9  0000G   CF  D0 0004D    MOVL    REAL_HOMEBLOCK, BUFFER+4                ; 0741
                                              08    A9  0000G   CF  D0 00053    MOVL    IDXHDR2_LBN, BUFFER+8                   ; 0742
                                              0C    A9  0201    8F  B0 00059    MOVW    #513, BUFFER+12                         ; 0743
                                                    56  0000G   CF  D0 0005F    MOVL    CLUSTER, R6                             ; 0745
                                              0E    A9          56  B0 00064    MOVW    R6, BUFFER+14                           ;
                                              10    A9          02  B0 00068    MOVW    #2, BUFFER+16                           ; 0746
                                50          0000G   CF  0000G   CF  C3 0006C    SUBL3   HOMEBLOCK2_LBN, REAL_HOMEBLOCK, R0      ; 0747
                                                    51      01 A046 3E 00074    MOVAW   1(R0)[R6], R1                           ;
                                              12    A9          51  B0 00079    MOVW    R1, BUFFER+18                           ;
                                52                  56          03  C5 0007D    MULL3   #3, R6, R2                              ; 0748
                            14  A9                  52          01  A1 00081    ADDW3   #1, R2, BUFFER+20                       ;
                                50                  56          02  78 00086    ASHL    #2, R6, R0                              ; 0749
```

```
              16   A9               50     01  A1 0008A         ADDW3   #1, R0, BUFFER+22                         ;
                             18     A9       0000G  CF  D0 0008F         MOVL    IDXFILE_LBN, BUFFER+24                    ; 0750
                             1C     A9       0000G  CF  D0 00095         MOVL    MAXIMUM, BUFFER+28                        ; 0751
                   50      0000G     CF  00000FFF  8F  C1 0009B         ADDL3   #4095, MAXIMUM, R0                        ; 0752
                   51                50  00001000  8F  C7 000A5         DIVL3   #4096, R0, R1                             ;
                             20     A9               51  B0 000AD         MOVW    R1, BUFFER+32                             ;
                             22     A9               09  B0 000B1         MOVW    #9, BUFFER+34                             ; 0753
                             2C     A9       0000G  CF  D0 000B5         MOVL    OWNER_UIC, BUFFER+44                      ; 0754
                             34     A9       0000G  CF  B0 000BB         MOVW    PROTECTION, BUFFER+52                     ; 0755
                                             0000G  CF  95 000C1         TSTB    INIT_OPTIONS                              ; 0756
                                                     04  18 000C5         BGEQ    1$                                        ;
                             2A     A9               01  88 000C7         BISB2   #1, BUFFER+42                             ; 0757
                                    04       0000G  CF  E9 000CB 1$:     BLBC    INIT_OPTIONS+1, 2$                        ; 0758
                             2A     A9               02  88 000D0         BISB2   #2, BUFFER+42                             ; 0759
                             36     A9       0000G  CF  B0 000D4 2$:     MOVW    FILE_PROT, BUFFER+54                      ; 0760
                             38     A9               6A  B0 000DA         MOVW    DEF_REC_PROT, BUFFER+56                   ; 0761
                             3C     A9               6E  7D 000DE         MOVQ    DATE_TIME, BUFFER+60                      ; 0762
                             44     A9       0000G  CF  90 000E2         MOVB    WINDOW, BUFFER+68                         ; 0764
                             45     A9       0000G  CF  90 000E8         MOVB    ACCESSED, BUFFER+69                       ; 0765
                             46     A9       0000G  CF  B0 000EE         MOVW    EXTENSION, BUFFER+70                      ; 0766
                           01C8     C9       0000G  CF  D0 000F4         MOVL    SERIAL_NUMBER, BUFFER+456                 ; 0767
                   04      0000G     CF               02  E1 000FB         BBC     #2, INIT_OPTIONS+5, 3$                    ; 0768
                             2A     A9               04  88 00101         BISB2   #4, BUFFER+42                             ; 0769
                   04      0000G     CF               03  E1 00105 3$:     BBC     #3, INIT_OPTIONS+5, 4$                    ; 0770
                             2A     A9               08  88 0010B         BISB2   #8, BUFFER+42                             ; 0771
        0C         20                6E               00  2C 0010F 4$:     MOVC5   #0, (SP), #32, #12, BUFFER+460            ; 0773
                                             01CC   C9      00114                                                   ;
        0C         20      0000G     DF      0000G  CF  2C 00117         MOVC5   LABEL_STRING, @LABEL_STRING+4, #32, #12, - ; 0775
                                             01D8   C9      00120                 BUFFER+472                                ;
        0C         20      0000G     DF      0000G  CF  2C 00123         MOVC5   USER_NAME, @USER_NAME+4, #32, #12, -      ; 0777
                                             01E4   C9      0012C                 BUFFER+484                                ;
              01F0   C9      FF2A     CA               0C  28 0012F         MOVC3   #12, FORMAT_NAME, BUFFER+496              ; 0778
                                      52               56  D0 00137         MOVL    R6, J                                     ; 0780
                                                       05  11 0013A         BRB     6$                                        ;
                           0000V     CF               00  FB 0013C 5$:     CALLS   #0, WRITE_HOMEBLOCK                       ; 0781
                                      F8               52  F5 00141 6$:     SOBGTR  J, 5$                                     ;
                                      69       0000G  CF  D0 00144         MOVL    HOMEBLOCK1_LBN, BUFFER                    ; 0783
                   52      0000G     CF               01  C1 00149         ADDL3   #1, CLUSTER, J                            ; 0784
                                                       05  11 0014F         BRB     8$                                        ;
                           0000V     CF               00  FB 00151 7$:     CALLS   #0, WRITE_HOMEBLOCK                       ; 0785
                                      F8               52  F5 00156 8$:     SOBGTR  J, 7$                                     ;
                                      69       0000G  CF  D0 00159         MOVL    HOMEBLOCK2_LBN, BUFFER                    ; 0787
                   52      0000G     CF               01  C1 0015E         ADDL3   #1, CLUSTER, J                            ; 0788
                                                       05  11 00164         BRB     10$                                       ;
                           0000V     CF               00  FB 00166 9$:     CALLS   #0, WRITE_HOMEBLOCK                       ; 0789
                                      F8               52  F5 0016B 10$:    SOBGTR  J, 9$                                     ;
    0200   8F              00         6E               00  2C 0016E         MOVC5   #0, (SP), #0, #512, BUFFER                ; 0795
                                                       69      00175                                                   ;
                                      69       01FF   8F  3C 00176         MOVZWL  #511, BUFFER                              ; 0796
                                      57       0000G  CF  D0 0017B         MOVL    IDXFILE_LBN, LBN                          ; 0797
                                               0280   8F  BB 00180         PUSHR   #^M<R7,R9>                                ; 0798
                                      6B               02  FB 00184         CALLS   #2, WRITE_BLOCK                           ;
                                                       69  D4 00187         CLRL    BUFFER                                    ; 0800
                   52      0000G     CF  00000FFF  8F  C1 00189         ADDL3   #4095, MAXIMUM, R2                        ; 0801
                                      52  00001000  8F  C6 00193         DIVL2   #4096, R2                                 ;
                                                       09  11 0019A         BRB     12$                                       ;
                                                       57  D6 0019C 11$:    INCL    LBN                                       ; 0803
```

```
                                                0280   8F   BB 0019E          PUSHR   #^M<R7,R9>                                                      ; 0804
                                           6B           02   FB 001A2          CALLS   #2, WRITE_BLOCK
                                           F4           52   F5 001A5 12$:     SOBGTR  J, 11$                                                          ; 0801
0200   8F                00      FF38      CA     00C8  8F   2C 001A8          MOVC5   #200, INITIAL_HEADER, #0, #512, BUFFER                          ; 0810
                                                      69   001B3
                                     3C    A9    0000G  CF   D0 001B4          MOVL    OWNER_UIC, BUFFER+60                                            ; 0812
                                     40    A9    0000G  CF   B0 001BA          MOVW    FILE_PROT, BUFFER+64                                            ; 0813
                                     38    A9           6A   B0 001C0          MOVW    DEF_REC_PROT, BUFFER+56                                         ; 0814
                                     66    A9           6E   7D 001C4          MOVQ    DATE_TIME, IDENT_AREA+22                                        ; 0815
                                     6E    A9           6E   7D 001C8          MOVQ    DATE_TIME, IDENT_AREA+30                                        ; 0817
                                           7E     01FE  8F   3C 001CC          MOVZWL  #510, -(SP)                                                     ; 0819
                                                      59   DD 001D1          PUSHL   R9
                                    0000G  CF           02   FB 001D3          CALLS   #2, CHECKSUM2
                                                      59   DD 001D8          PUSHL   R9                                                              ; 0820
                                                05    A7   9F 001DA          PUSHAB  5(LBN)
                                           6B           02   FB 001DD          CALLS   #2, WRITE_BLOCK
                                     08    A9 00070007  8F   D0 001E0          MOVL    #458759, BUFFER+8                                               ; 0825
                          50   A9       04    AA           06   28 001E8          MOVC3   #6, P.AAF, IDENT_AREA                                           ; 0827
                                           7E     01FE  8F   3C 001EE          MOVZWL  #510, -(SP)                                                     ; 0828
                                                      59   DD 001F3          PUSHL   R9
                                    0000G  CF           02   FB 001F5          CALLS   #2, CHECKSUM2
                                                      59   DD 001FA          PUSHL   R9                                                              ; 0829
                                                07    A7   9F 001FC          PUSHAB  7(LBN)
                                           6B           02   FB 001FF          CALLS   #2, WRITE_BLOCK
                                     08    A9 00060006  8F   D0 00202          MOVL    #393222, BUFFER+8                                               ; 0834
                                     16    A9       40  8F   9B 0020A          MOVZBW  #64, BUFFER+22                                                  ; 0836
                                     24    A9       40  8F   9B 0020F          MOVZBW  #64, BUFFER+36                                                  ; 0837
                          50   A9       0A    AA           06   28 00214          MOVC3   #6, P.AAG, IDENT_AREA                                           ; 0838
                                           7E     01FE  8F   3C 0021A          MOVZWL  #510, -(SP)                                                     ; 0839
                                                      59   DD 0021F          PUSHL   R9
                                    0000G  CF           02   FB 00221          CALLS   #2, CHECKSUM2
                                                      59   DD 00226          PUSHL   R9                                                              ; 0840
                                                06    A7   9F 00228          PUSHAB  6(LBN)
                                           6B           02   FB 0022B          CALLS   #2, WRITE_BLOCK
                                     08    A9 00080008  8F   D0 0022E          MOVL    #524296, BUFFER+8                                               ; 0845
                     50   A9          10    AA           06   28 00236          MOVC3   #6, P.AAH, IDENT_AREA                                           ; 0847
                                           7E     01FE  8F   3C 0023C          MOVZWL  #510, -(SP)                                                     ; 0848
                                                      59   DD 00241          PUSHL   R9
                                    0000G  CF           02   FB 00243          CALLS   #2, CHECKSUM2
                                                      59   DD 00248          PUSHL   R9                                                              ; 0849
                                                08    A7   9F 0024A          PUSHAB  8(LBN)
                                           6B           02   FB 0024D          CALLS   #2, WRITE_BLOCK
                                     08    A9 00090009  8F   D0 00250          MOVL    #589833, BUFFER+8                                               ; 0854
                                     16    A9           10   B0 00258          MOVW    #16, BUFFER+22                                                  ; 0856
                                     24    A9           10   B0 0025C          MOVW    #16, BUFFER+36                                                  ; 0857
                          50   A9       16    AA           06   28 00260          MOVC3   #6, P.AAI, IDENT_AREA                                           ; 0858
                                           7E     01FE  8F   3C 00266          MOVZWL  #510, -(SP)                                                     ; 0859
                                                      59   DD 0026B          PUSHL   R9
                                    0000G  CF           02   FB 0026D          CALLS   #2, CHECKSUM2
                                                      59   DD 00272          PUSHL   R9                                                              ; 0860
                                                09    A7   9F 00274          PUSHAB  9(LBN)
                                           6B           02   FB 00277          CALLS   #2, WRITE_BLOCK
                                     08    A9 00010001  8F   D0 0027A          MOVL    #65537, BUFFER+8                                                ; 0865
                                     50       0000G     CF   D0 00282          MOVL    CLUSTER, R0                                                     ; 0867
                                     4C    A9    0000GDF40   DE 00287          MOVAL   @IDXFILE_CNT[R0], BUFFER+76
                                              4C    A9     D6 0028E          INCL    BUFFER+76
                                     16    A9       0200  8F   B0 00291          MOVW    #512, BUFFER+22                                                 ; 0868
```

```
                    24    A9    0200    8F   B0 00297          MOVW    #512, BUFFER+36                          ; 0869
                          51    0000GDF40   DE 0029D          MOVAL   @IDXFILE_CNT[R0], R1                     ; 0870
              18    A9          51          10   9C 002A3     ROTL    #16, R1, BUFFER+24                       ;
              51          0000G CF 00000FFF 8F   C1 002A8     ADDL3   #4095, MAXIMUM, R1                       ; 0871
                          51 00001000 8F   C6 002B2          DIVL2   #4096, R1                                ;
                          50      0A A140   DE 002B9         MOVAL   10(R1)[R0], R0                           ;
              1C    A9          50          10   9C 002BE     ROTL    #16, R0, BUFFER+28                       ;
              50    A9    1C    AA          06   28 002C3     MOVC3   #6, P.AAJ, IDENT_AREA                    ; 0872
                          56    0000G   CF    D0 002C9         MOVL    BOOTBLOCK_CNT, MAP_COUNT                 ; 0873
                          53    0000G   CF    D0 002CE         MOVL    BOOTBLOCK_LBN, MAP_LBN                   ; 0874
                    52 00000000G  8F          01   C3 002D3     SUBL3   #1, #BOOTBLOCK_IDX+1, J                  ; 0875
                                              29   11 002DB     BRB     15$                                      ;
                    50          56          53   C1 002DD 13$:  ADDL3   MAP_LBN, MAP_COUNT, R0                  ; 0877
                          0000GCF42           50   D1 002E1     CMPL    R0, ALLOC_TABLE_LBN[J]                   ;
                                              08   12 002E7     BNEQ    14$                                      ;
                          56      0000GCF42   C0 002E9         ADDL2   ALLOC_TABLE_CNT[J], MAP_COUNT            ; 0879
                                              15   11 002EF     BRB     15$                                      ;
                                              53   DD 002F1 14$:  PUSHL   MAP_LBN                                 ; 0882
                                              56   DD 002F3     PUSHL   MAP_COUNT                                ;
                          0000V   CF          02   FB 002F5     CALLS   #2, MAKE_POINTER                         ;
                          56      0000GCF42   D0 002FA         MOVL    ALLOC_TABLE_CNT[J], MAP_COUNT            ; 0883
                          53      0000GCF42   D0 00300         MOVL    ALLOC_TABLE_LBN[J], MAP_LBN              ; 0884
                    D3          52          00G  F3 00306 15$:  AOBLEQ  S^IDXFILE_IDX, J, 13$                   ; 0875
                                              53   DD 0030A     PUSHL   MAP_LBN                                  ; 0887
                                              56   DD 0030C     PUSHL   MAP_COUNT                                ;
                          0000V   CF          02   FB 0030E     CALLS   #2, MAKE_POINTER                         ;
                                  7E    01FE  8F   3C 00313     MOVZWL  #510, -(SP)                              ; 0889
                                              59   DD 00318     PUSHL   R9                                       ;
                          0000G   CF          02   FB 0031A     CALLS   #2, CHECKSUM2                            ;
                                              59   DD 0031F     PUSHL   R9                                       ; 0890
                                        01    A7   9F 00321     PUSHAB  1(LBN)                                   ;
                                  6B          02   FB 00324     CALLS   #2, WRITE_BLOCK                          ;
                                  52      0000G CF   D0 00327     MOVL    CLUSTER, J                               ; 0891
                                              0A   11 0032C     BRB     17$                                      ;
                                              59   DD 0032E 16$:  PUSHL   R9                                      ; 0892
                                      0000GDF42    9F 00330     PUSHAB  @IDXHDR2_LBN[J]                          ;
                                  6B          02   FB 00335     CALLS   #2, WRITE_BLOCK                          ;
                                  F3          52   F4 00338 17$:  SOBGEQ  J, 16$                                ;
0138  8F          00              6E          00   2C 0033B     MOVC5   #0, (SP), #0, #312, BUFFER+200           ; 0897
                                      00C8    C9   00342                                                       ;
                                        3A    A9   94 00345     CLRB    BUFFER+58                                ; 0898
                          08    A9 00030003 8F   D0 00348     MOVL    #196611, BUFFER+8                        ; 0899
                                              56   D4 00350     CLRL    MAP_COUNT                                ; 0902
                                  58      0000G CF   D0 00352     MOVL    BADBLOCK_TOTAL, R8                       ; 0903
                                  50          01   CE 00357     MNEGL   #1, J                                    ;
                                              06   11 0035A     BRB     19$                                      ;
                                  56      0000GCF40    C0 0035C 18$:  ADDL2   BADBLOCK_CNT[J], MAP_COUNT       ; 0904
                    F6            50          58   F2 00362 19$:  AOBLSS  R8, J, 18$                           ;
                                  50      01   A6   9E 00366     MOVAB   1(R6), R0                                ; 0905
                          4C    A9          50   D0 0036A     MOVL    R0, BUFFER+76                            ;
              18    A9          56          10   9C 0036E     ROTL    #16, MAP_COUNT, BUFFER+24                ; 0906
              1C    A9          50          10   9C 00373     ROTL    #16, R0, BUFFER+28                       ; 0907
              50    A9    22    AA          06   28 00378     MOVC3   #6, P.AAK, IDENT_AREA                    ; 0909
                                  52          01   CE 0037E     MNEGL   #1, J                                    ; 0910
                                              23   11 00381     BRB     22$                                      ;
                          97    8F    3A    A9   91 00383 20$:  CMPB    BUFFER+58, #151                          ; 0912
                                              0D   1B 00388     BLEQU   21$                                      ;
```

```
                                                007580BC    8F   DD 0038A            PUSHL   #7700668                                    ; 0913
                                    00000000G   00           01   FB 00390            CALLS   #1, LIB$STOP
                                                0000GCF42    DD 00397 21$:          PUSHL   BADBLOCK_LBN[J]                             ; 0914
                                                0000GCF42    DD 0039C               PUSHL   BADBLOCK_CNT[J]
                                          0000V CF           02   FB 003A1            CALLS   #2, MAKE_POINTER
                                    D9          52           58   F2 003A6 22$:       AOBLSS  R8, J, 20$                                  ; 0910
                                                7E      01FE 8F   3C 003AA            MOVZWL  #510, -(SP)                                 ; 0916
                                                            59   DD 003AF            PUSHL   R9
                                          0000G CF           02   FB 003B1            CALLS   #2, CHECKSUM2
                                                            59   DD 003B6            PUSHL   R9                                          ; 0917
                                                         03 A7   9F 003B8            PUSHAB  3(LBN)
                                                6B          02   FB 003BB            CALLS   #2, WRITE_BLOCK
    0138    8F          00                      6E          00   2C 003BE            MOVC5   #0, (SP), #0, #312, BUFFER+200              ; 0922
                                                      00C8  C9      003C5
                                                         3A A9   94 003C8            CLRB    BUFFER+58                                   ; 0923
                                             08  A9 00020002 8F   D0 003CB            MOVL    #131074, BUFFER+8                           ; 0924
                                             34  A9       80 8F   88 003D3            BISB2   #128, BUFFER+52                             ; 0926
                                    50    0000G CF     0000G CF   C7 003D8            DIVL3   CLUSTER, VOLUME_SIZE, R0                    ; 0927
                                                50   0FFF C0   9E 003E0            MOVAB   4095(R0), R0
                                                50 00001000 8F   C6 003E5            DIVL2   #4096, R0
                                                50          02   C0 003EC            ADDL2   #2, R0
                                             4C  A9         50   D0 003EF            MOVL    R0, BUFFER+76
                                18  A9       0000G CF       10   9C 003F3            ROTL    #16, BITMAP_CNT, BUFFER+24                  ; 0928
                                1C  A9             50       10   9C 003FA            ROTL    #16, R0, BUFFER+28                          ; 0929
                                50  A9         28  AA       06   28 003FF            MOVC3   #6, P.AAL, IDENT_AREA                       ; 0931
                                                0000G CF     DD 00405               PUSHL   BITMAP_LBN                                  ; 0932
                                                0000G CF     DD 00409               PUSHL   BITMAP_CNT
                                          0000V CF           02   FB 0040D            CALLS   #2, MAKE_POINTER
                                                7E      01FE 8F   3C 00412            MOVZWL  #510, -(SP)                                 ; 0933
                                                            59   DD 00417            PUSHL   R9
                                          0000G CF           02   FB 00419            CALLS   #2, CHECKSUM2
                                                            59   DD 0041E            PUSHL   R9                                          ; 0934
                                                         02 A7   9F 00420            PUSHAB  2(LBN)
                                                6B          02   FB 00423            CALLS   #2, WRITE_BLOCK
    0138    8F          00                      6E          00   2C 00426            MOVC5   #0, (SP), #0, #312, BUFFER+200              ; 0939
                                                      00C8  C9      0042D
                                                         3A A9   94 00430            CLRB    BUFFER+58                                   ; 0940
                                             08  A9 00040004 8F   D0 00433            MOVL    #262148, BUFFER+8                           ; 0941
                                             35  A9       20 88   0043B            BISB2   #32, BUFFER+53                              ; 0943
                                             40  A9     4444 8F   AA 0043F            BICW2   #17476, BUFFER+64                           ; 0944
                                             4C  A9          02   D0 00445            MOVL    #2, BUFFER+76                               ; 0945
                                             1C  A9 00020000 8F   D0 00449            MOVL    #131072, BUFFER+28                          ; 0946
                                18  A9       0000G CF       10   9C 00451            ROTL    #16, MFD_CNT, BUFFER+24                     ; 0947
                                             14  A9     0802 8F   B0 00458            MOVW    #2050, BUFFER+20                            ; 0948
                                50  A9         2E  AA       0A   28 0045E            MOVC3   #10, P.AAM, IDENT_AREA                      ; 0951
                                                0000G CF     DD 00464               PUSHL   MFD_LBN                                     ; 0952
                                                0000G CF     DD 00468               PUSHL   MFD_CNT
                                          0000V CF           02   FB 0046C            CALLS   #2, MAKE_POINTER
                                                7E      01FE 8F   3C 00471            MOVZWL  #510, -(SP)                                 ; 0953
                                                            59   DD 00476            PUSHL   R9
                                          0000G CF           02   FB 00478            CALLS   #2, CHECKSUM2
                                                            59   DD 0047D            PUSHL   R9                                          ; 0954
                                                         04 A7   9F 0047F            PUSHAB  4(LBN)
                                                6B          02   FB 00482            CALLS   #2, WRITE_BLOCK
                                                            04   00485            RET                                                 ; 0956

; Routine Size:  1158 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  539          0957  1 ROUTINE WRITE_HOMEBLOCK : NOVALUE =
;  540          0958  1
;  541          0959  1 !++
;  542          0960  1 !
;  543          0961  1 ! FUNCTIONAL DESCRIPTION:
;  544          0962  1 !
;  545          0963  1 !       This routine computes the checksums in the home block currently
;  546          0964  1 !       in the buffer, writes it, and then increments the block numbers
;  547          0965  1 !       in the home block for the next write.
;  548          0966  1 !
;  549          0967  1 !
;  550          0968  1 ! CALLING SEQUENCE:
;  551          0969  1 !       WRITE_HOMEBLOCK ()
;  552          0970  1 !
;  553          0971  1 ! INPUT PARAMETERS:
;  554          0972  1 !       NONE
;  555          0973  1 !
;  556          0974  1 ! IMPLICIT INPUTS:
;  557          0975  1 !       BUFFER contains home block
;  558          0976  1 !
;  559          0977  1 ! OUTPUT PARAMETERS:
;  560          0978  1 !       NONE
;  561          0979  1 !
;  562          0980  1 ! IMPLICIT OUTPUTS:
;  563          0981  1 !       NONE
;  564          0982  1 !
;  565          0983  1 ! ROUTINE VALUE:
;  566          0984  1 !       NONE
;  567          0985  1 !
;  568          0986  1 ! SIDE EFFECTS:
;  569          0987  1 !       home block written
;  570          0988  1 !
;  571          0989  1 !--
;  572          0990  1
;  573          0991  2 BEGIN
;  574          0992  2
;  575          0993  2 EXTERNAL
;  576          0994  2         BUFFER          : BBLOCK;       ! buffer containing home block
;  577          0995  2
;  578          0996  2 EXTERNAL ROUTINE
;  579          0997  2         CHECKSUM2,                      ! block checksum routine
;  580          0998  2         WRITE_BLOCK;                    ! write a block to the disk
;  581          0999  2
;  582          1000  2
;  583          1001  2 ! Compute the two checksums and then write the block.
;  584          1002  2 !
;  585          1003  2
;  586          1004  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (HM2$W_CHECKSUM1));
;  587          1005  2 CHECKSUM2 (BUFFER, $BYTEOFFSET (HM2$W_CHECKSUM2));
;  588          1006  2 WRITE_BLOCK (.BUFFER[HM2$L_HOMELBN], BUFFER);
;  589          1007  2
;  590          1008  2 ! Advance the block numbers to those of the next home block.
;  591          1009  2 !
;  592          1010  2
;  593          1011  2 BUFFER[HM2$L_HOMELBN] = .BUFFER[HM2$L_HOMELBN] + 1;
;  594          1012  2 BUFFER[HM2$W_HOMEVBN] = .BUFFER[HM2$W_HOMEVBN] + 1;
;  595          1013  2
```

```
;  596          1014  1 END;                       ! end of routine WRITE_HOMEBLOCK



                                        0004 00000 WRITE_HOMEBLOCK:
                                                   .WORD   Save R2                                          ; 0957
                        52      0000G   CF  9E 00002      MOVAB   BUFFER, R2                               ;
                                        3A  DD 00007      PUSHL   #58                                      ; 1004
                                        52  DD 00009      PUSHL   R2                                       ;
                0000G   CF              02  FB 0000B      CALLS   #2, CHECKSUM2                            ;
                        7E      01FE    8F  3C 00010      MOVZWL  #510, -(SP)                              ; 1005
                                        52  DD 00015      PUSHL   R2                                       ;
                0000G   CF              02  FB 00017      CALLS   #2, CHECKSUM2                            ;
                                        52  DD 0001C      PUSHL   R2                                       ; 1006
                                        62  DD 0001E      PUSHL   BUFFER                                   ;
                0000G   CF              02  FB 00020      CALLS   #2, WRITE_BLOCK                          ;
                                        62  D6 00025      INCL    BUFFER                                   ; 1011
                                10      A2  B6 00027      INCW    BUFFER+16                                ; 1012
                                            04 0002A      RET                                              ; 1014

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0486
```

```
;   598          1015  1 ROUTINE MAKE_POINTER (COUNT, LBN) : NOVALUE =
;   599          1016  1
;   600          1017  1 !++
;   601          1018  1 !
;   602          1019  1 ! FUNCTIONAL DESCRIPTION:
;   603          1020  1 !
;   604          1021  1 !       This routine appends a retrieval pointer to the map area of the current
;   605          1022  1 !       file header describing the given count and LBN.
;   606          1023  1 !
;   607          1024  1 !
;   608          1025  1 ! CALLING SEQUENCE:
;   609          1026  1 !       MAKE_POINTER (ARG1, ARG2)
;   610          1027  1 !
;   611          1028  1 ! INPUT PARAMETERS:
;   612          1029  1 !       ARG1: block count
;   613          1030  1 !       ARG2: start LBN
;   614          1031  1 !
;   615          1032  1 ! IMPLICIT INPUTS:
;   616          1033  1 !       BUFFER contains file header
;   617          1034  1 !
;   618          1035  1 ! OUTPUT PARAMETERS:
;   619          1036  1 !       NONE
;   620          1037  1 !
;   621          1038  1 ! IMPLICIT OUTPUTS:
;   622          1039  1 !       retrieval pointer added to header
;   623          1040  1 !
;   624          1041  1 ! ROUTINE VALUE:
;   625          1042  1 !       NONE
;   626          1043  1 !
;   627          1044  1 ! SIDE EFFECTS:
;   628          1045  1 !       NONE
;   629          1046  1 !
;   630          1047  1 !--
;   631          1048  1
;   632          1049  2 BEGIN
;   633          1050  2
;   634          1051  2 BUILTIN
;   635          1052  2         ROT;
;   636          1053  2
;   637          1054  2 LOCAL
;   638          1055  2         MAP_POINTER     : REF BBLOCK;   ! pointer to map area
;   639          1056  2
;   640          1057  2 EXTERNAL
;   641          1058  2         BUFFER          : BBLOCK;       ! buffer containing file header
;   642          1059  2
;   643          1060  2
;   644          1061  2 ! Compute the address in the file header where the pointer should go.
;   645          1062  2 ! Then determine the format of the pointer and build it.
;   646          1063  2 !
;   647          1064  2
;   648          1065  2 MAP_POINTER = BUFFER + 2 * (.BUFFER[FH2$B_MPOFFSET] + .BUFFER[FH2$B_MAP_INUSE]);
;   649          1066  2
;   650          1067  2 IF .COUNT LEQU 256 AND .LBN LSSU 1^22
;   651          1068  2 THEN
;   652          1069  3     BEGIN
;   653          1070  3     MAP_POINTER[FM2$V_FORMAT] = FM2$C_FORMAT1;
;   654          1071  3     MAP_POINTER[FM2$B_COUNT1] = .COUNT - 1;
```

```
;  655          1072  3      MAP_POINTER[FM2$V_HIGHLBN] = .LBN<16,6>;
;  656          1073  3      MAP_POINTER[FM2$W_LOWLBN] = .LBN<0,16>;
;  657          1074  3      BUFFER[FH2$B_MAP_INUSE] = .BUFFER[FH2$B_MAP_INUSE] + 2;
;  658          1075  3      END
;  659          1076  3
;  660          1077  2  ELSE IF .COUNT LEQU 16384
;  661          1078  2  THEN
;  662          1079  3      BEGIN
;  663          1080  3      MAP_POINTER[FM2$V_FORMAT] = FM2$C_FORMAT2;
;  664          1081  3      MAP_POINTER[FM2$V_COUNT2] = .COUNT - 1;
;  665          1082  3      MAP_POINTER[FM2$L_LBN2] = .LBN;
;  666          1083  3      BUFFER[FH2$B_MAP_INUSE] = .BUFFER[FH2$B_MAP_INUSE] + 3;
;  667          1084  3      END
;  668          1085  3
;  669          1086  2  ELSE IF .COUNT LEQU 1^30
;  670          1087  2  THEN
;  671          1088  3      BEGIN
;  672          1089  3      .MAP_POINTER = ROT (.COUNT-1, 16);
;  673          1090  3      MAP_POINTER[FM2$V_FORMAT] = FM2$C_FORMAT3;
;  674          1091  3      MAP_POINTER[FM2$L_LBN3] = .LBN;
;  675          1092  3      BUFFER[FH2$B_MAP_INUSE] = .BUFFER[FH2$B_MAP_INUSE] + 4;
;  676          1093  3      END
;  677          1094  3
;  678          1095  2  ELSE ERR_EXIT (INIT$_LARGECNT);
;  679          1096  2
;  680          1097  1  END;                                    ! end of routine MAKE_POINTER


                                        000C 00000 MAKE_POINTER:
                                                    .WORD   Save R2,R3                          ;  1015
                                53      0000G   CF  9E 00002    MOVAB   BUFFER+58, R3           ;  1065
                                50        C7    A3  9A 00007    MOVZBL  BUFFER+1, R0            ;
                                51              63  9A 0000B    MOVZBL  BUFFER+58, R1           ;
                                50              51  C0 0000E    ADDL2   R1, R0                  ;
                                50        C6 A340   3E 00011    MOVAW   BUFFER[R0], MAP_POINTER ;
                                51        04    AC  D0 00016    MOVL    COUNT, R1               ;  1067
                    00000100    8F              51  D1 0001A    CMPL    R1, #256                ;
                                                23  1A 00021    BGTRU   1$                      ;
                    00400000    8F        08    AC  D1 00023    CMPL    LBN, #4194304           ;
                                                19  1E 0002B    BGEQU   1$                      ;
          60        02                    0E      01  F0 0002D    INSV    #1, #14, #2, (MAP_POINTER)  ;  1070
          60                              51      01  83 00032    SUBB3   #1, R1, (MAP_POINTER)   ;  1071
01        A0        06                    00  0A  AC  F0 00036    INSV    LBN+2, #0, #6, 1(MAP_POINTER)  ;  1072
                          02              A0  08  AC  B0 0003D    MOVW    LBN, 2(MAP_POINTER)     ;  1073
                                63        02      80  00042    ADDB2   #2, BUFFER+58           ;  1074
                                                04  00045    RET                             ;  1067
                    00004000    8F              51  D1 00046 1$:    CMPL    R1, #16384          ;  1077
                                                17  1A 0004D    BGTRU   2$                      ;
          60        02                    0E      02  F0 0004F    INSV    #2, #14, #2, (MAP_POINTER)  ;  1080
                                52        FF    A1  9E 00054    MOVAB   -1(R1), R2              ;  1081
          60        0E                    00      52  F0 00058    INSV    R2, #0, #14, (MAP_POINTER)  ;
                          02              A0  08  AC  D0 0005D    MOVL    LBN, 2(MAP_POINTER)     ;  1082
                                63        03      80  00062    ADDB2   #3, BUFFER+58           ;  1083
                                                04  00065    RET                             ;  1077
```

```
                          40000000   8F              51  D1 00066 2$:     CMPL    R1, #1073741824                         ; 1086
                                                     14  1A 0006D        BGTRU   3$                                      ;
                                                     51  D7 0006F        DECL    R1                                      ; 1089
                      60             51              10  9C 00071        ROTL    #16, R1, (MAP_POINTER)                  ;
                            01       A0        C0    8F  88 00075        BISB2   #192, 1(MAP_POINTER)                    ; 1090
                            04       A0        08    AC  D0 0007A        MOVL    LBN, 4(MAP_POINTER)                     ; 1091
                                     63              04  80 0007F        ADDB2   #4, BUFFER+58                           ; 1092
                                                     04  00082        RET                                             ; 1086
                                007580DC   8F          DD  00083 3$:     PUSHL   #7700700                                ; 1095
                          00000000G  00                01  FB 00089        CALLS   #1, LIB$STOP                            ;
                                                     04  00090        RET                                             ; 1097

; Routine Size:  145 bytes,    Routine Base:  $CODE$ + 04B1

;  681          1098  1
;  682          1099  1 END
;  683          1100  0 ELUDOM



                                                                    .EXTRN  LIB$STOP

;                                  PSECT SUMMARY
;
;       Name                     Bytes                          Attributes
;
;  $PLIT$                          348  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         1346  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        86         0        1000          00:01.9



;                                     COMMAND QUALIFIERS

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

; Size:         1346 code + 348 data bytes
; Run Time:       00:31.5
; Elapsed Time:   01:05.2
; Lines/CPU Min:   2092
; Lexemes/CPU-Min: 28251
; Memory Used:  324 pages
```

```
; Compilation Complete
```