```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC  KKK          KKK  UUUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC  KKK          KKK  UUUUUUUUUUUUUUUU  PPP
```

```
  SSSSSSSS  TTTTTTTTTT    AAAAAA      IIIIII    NN      NN    IIIIII    VV      VV    000000    LL
  SSSSSSSS  TTTTTTTTTT    AAAAAA      IIIIII    NN      NN    IIIIII    VV      VV    000000    LL
SS              TT       AA    AA      II      NN      NN      II      VV      VV   00    00   LL
SS              TT       AA    AA      II      NN      NN      II      VV      VV   00    00   LL
SS              TT       AA    AA      II      NNNN    NN      II      VV      VV   00    00   LL
SS              TT       AA    AA      II      NNNN    NN      II      VV      VV   00    00   LL
  SSSSSS        TT       AA    AA      II      NN  NN  NN      II      VV      VV   00    00   LL
  SSSSSS        TT       AA    AA      II      NN  NN  NN      II      VV      VV   00    00   LL
        SS      TT       AAAAAAAAAA    II      NN    NNNN      II      VV      VV   00    00   LL
        SS      TT       AAAAAAAAAA    II      NN    NNNN      II      VV      VV   00    00   LL
        SS      TT       AA    AA      II      NN      NN      II        VV  VV     00    00   LL
        SS      TT       AA    AA      II      NN      NN      II        VV  VV     00    00   LL
SSSSSSSS        TT       AA    AA    IIIIII    NN      NN    IIIIII        VV       000000    LLLLLLLLLL
SSSSSSSS        TT       AA    AA    IIIIII    NN      NN    IIIIII        VV       000000    LLLLLLLLLL

LL              IIIIII     SSSSSSSS
LL              IIIIII     SSSSSSSS
LL                II     SS
LL                II     SS
LL                II     SS
LL                II     SS
LL                II       SSSSSS
LL                II       SSSSSS
LL                II             SS
LL                II             SS
LL                II             SS
LL                II             SS
LLLLLLLLLL      IIIIII     SSSSSSSS
LLLLLLLLLL      IIIIII     SSSSSSSS
```

```
;     1          0001  0 MODULE STAINIVOL(%TITLE 'Disk volume initialization'
;     2          0002  0                   IDENT = 'V04-000'
;     3          0003  0                   ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !**********************************************************************
;     8          0008  1 !*                                                                    *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                            *
;    12          0012  1 !*                                                                    *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    14          0014  1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  *
;    18          0018  1 !*  TRANSFERRED.                                                    *
;    19          0019  1 !*                                                                    *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  *
;    21          0021  1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  *
;    22          0022  1 !*  CORPORATION.                                                    *
;    23          0023  1 !*                                                                    *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.         *
;    26          0026  1 !*                                                                    *
;    27          0027  1 !*                                                                    *
;    28          0028  1 !**********************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:
;    33          0033  1 !       Backup/Restore
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !       This module contains the routines to initialize a disk volume.
;    37          0037  1 !       These routines are adapted from and must track the INIT utility.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !       VAX/VMS user mode.
;    41          0041  1 !--
;    42          0042  1 !
;    43          0043  1 ! AUTHOR: M. Jack, CREATION DATE: 12-Feb-1980
;    44          0044  1 !
;    45          0045  1 ! MODIFIED BY:
;    46          0046  1 !
;    47          0047  1 !       V03-006 ACG0332         Andrew C. Goldstein,    29-Apr-1983  15:48
;    48          0048  1 !               Add full highwater mark support
;    49          0049  1 !
;    50          0050  1 !       V03-005 ACG0325         Andrew C. Goldstein,    4-Apr-1983  16:04
;    51          0051  1 !               Fix use of file header length symbol; add file
;    52          0052  1 !               name extension.
;    53          0053  1 !
;    54          0054  1 !       V03-004 ACG0324         Andrew C. Goldstein,    1-Apr-1983  17:51
;    55          0055  1 !               Set storage map size in VCB
;    56          0056  1 !
;    57          0057  1 !       V03-003 ACG0313         Andrew C. Goldstein,    12-Feb-1983  17:17
```

```
;    58          0058  1 !                Add routine subtitles
;    59          0059  1 !
;    60          0060  1 !        V03-002 MLJ48676        Martin L. Jack, 6-Sep-1982  21:04
;    61          0061  1 !                Set the index file EFBLK at the highest used file header.
;    62          0062  1 !
;    63          0063  1 !        V03-001 LMP0022         L. Mark Pilant,         5-Apr-1982  17:10
;    64          0064  1 !                Add support for multi-header index files.
;    65          0065  1 !
;    66          0066  1 !        V02-003 MLJ0081         Martin L. Jack, 26-Feb-1982  15:55
;    67          0067  1 !                Implement RETAINMIN and RETAINMAX for new home block fields.
;    68          0068  1 !
;    69          0069  1 !        V02-002 MLJ0054         Martin L. Jack, 22-Nov-1981  22:46
;    70          0070  1 !                Integrate GET_VM and FREE_VM jacket routines.
;    71          0071  1 !
;    72          0072  1 !        V02-001 ACG0211         Andrew C. Goldstein,    16-Jul-1981  10:47
;    73          0073  1 !                Add logic to initialize save set disks
;    74          0074  1 !
;    75          0075  1 !**
```

```
;    77          0076  1 REQUIRE 'SRC$:COMMON';
;    78          1182  1 LIBRARY 'SYS$LIBRARY:LIB';
;    79          1183  1
;    80          1184  1
;    81          1185  1 LINKAGE
;    82          1186  1         L_P$=           CALL: GLOBAL(P$=11);
;    83          1187  1
;    84          1188  1
;    85          1189  1 MACRO
;    86          1190  1         L_DECL=         EXTERNAL REGISTER P$ = 11: REF VECTOR %;
;    87          1191  1
;    88          1192  1
;    89          1193  1 FORWARD ROUTINE
;    90          1194  1         ALLOCATE:       L_P$ NOVALUE,   ! Allocate blocks
;    91          1195  1         ALLOCATE_HOME:  L_P$ NOVALUE,   ! Allocate on home block sequence
;    92          1196  1         INIT_ALLOCATE:  L_P$ NOVALUE,   ! Allocate file structure
;    93          1197  1         WRITE_BLOCK:    L_P$ NOVALUE,   ! Write block by LBN
;    94          1198  1         INIT_BITMAP:    L_P$ NOVALUE,   ! Initialize bitmap file
;    95          1199  1         INIT_INDEX1:    L_P$ NOVALUE,   ! Initialize index file (ODS-1)
;    96          1200  1         INIT_INDEX:     L_P$ NOVALUE,   ! Initialize index file (ODS-2)
;    97          1201  1         INIT_MFD:       L_P$ NOVALUE,   ! Initialize MFD (ODS-2)
;    98          1202  1         WRITE_HOMEBLOCK:L_P$ NOVALUE,   ! Write home block (ODS-2)
;    99          1203  1         INITIALIZE_VOLUME:
;   100          1204  1                         NOVALUE;        ! Driver for initialization
;   101          1205  1
;   102          1206  1
;   103          1207  1 EXTERNAL ROUTINE
;   104          1208  1         FREE_VM:        NOVALUE,        ! Free virtual memory
;   105          1209  1         GET_VM,                         ! Allocate virtual memory
;   106          1210  1         GET_ZERO_VM,                    ! Allocate and clear virtual memory
;   107          1211  1         CHECKSUM,                       ! Compute header checksum
;   108          1212  1         CHECKSUM2,                      ! Compute block checksum
;   109          1213  1         CREATE_WINDOW,                  ! Create window control block
;   110          1214  1         GET_BADBLOCKS,                  ! Get bad block data
;   111          1215  1         MAKE_POINTER1,                  ! Make map pointer (ODS-1)
;   112          1216  1         MAKE_POINTER,                   ! Make map pointer (ODS-2)
;   113          1217  1         STA_ALLOC_LBN,                  ! Allocate by LBN
;   114          1218  1         TO_ODS1_DATE:   NOVALUE;        ! Convert 64-bit time to ODS-1 format
;   115          1219  1
;   116          1220  1
;   117          1221  1 EXTERNAL LITERAL
;   118          1222  1         BACKUP$_ALLOCFAIL,
;   119          1223  1         BACKUP$_BLKZERO,
;   120          1224  1         BACKUP$_CLUSTER,
;   121          1225  1         BACKUP$_LARGECNT,
;   122          1226  1         BACKUP$_MAXBAD,
;   123          1227  1         BACKUP$_WRITEERR;
;   124          1228  1
;   125          1229  1
;   126          1230  1 G$DEFINE();                           ! Define global area
;   127          1231  1
;   128          1232  1
;   129          1233  1 BUILTIN
;   130          1234  1         ROT;
```

```
;  132          1235  1 %SBTTL 'Allocation table'
;  133          1236  1 LITERAL
;  134          1237  1         ALLOC_MAX=      7;              ! Size of allocation table
;  135          1238  1
;  136          1239  1
;  137          1240  1 MACRO
;  138          1241  1         CHANNEL=        P$[0] %,        ! Channel number
;  139          1242  1         STRUCLEV_1=     P$[1] %,        ! True if ODS-1
;  140          1243  1         HOMEBLOCK_DELTA=P$[2] %,        ! Home block delta factor
;  141          1244  1         VOLUME_SIZE=    P$[3] %,        ! Size rounded to cluster boundary
;  142          1245  1         REAL_HOMEBLOCK= P$[4] %,        ! LBN of actual secondary home block
;  143          1246  1         CLUSTER=        P$[5] %,        ! Cluster factor
;  144          1247  1         _VCB=           P$[6] %,        ! Pointer to VCB for the volume
;  145          1248  1         _DEVCHAR=       P$[7] %,        ! Pointer to device characteristics
;  146          1249  1         VCB(O,P,S,E)=           BBLOCK[._VCB,O,P,S,E] %,
;  147          1250  1         DEVCHAR(O,P,S,E)=       BBLOCK[._DEVCHAR,O,P,S,E] %,
;  148          1251  1
;  149          1252  1
;  150          1253  1         ! Allocation table.  Consists of 2 parallel tables, for size and LBN of
;  151          1254  1         ! allocated areas.  Each contains an entry for each piece of the disk
;  152          1255  1         ! that is allocated to something.
;  153          1256  1         !
;  154          1257  1         _ALLOC_TABLE_CNT=P$[8] %,
;  155          1258  1         _ALLOC_TABLE_LBN=P$[8+ALLOC_MAX] %,
;  156          1259  1         ALLOC_TABLE_CNT(N)=     VECTOR[_ALLOC_TABLE_CNT,N] %,
;  157          1260  1         ALLOC_TABLE_LBN(N)=     VECTOR[_ALLOC_TABLE_LBN,N] %;
;  158          1261  1
;  159          1262  1
;  160          1263  1 LITERAL
;  161          1264  1         P$SIZE=         8 + ALLOC_MAX + ALLOC_MAX;
;  162          1265  1
;  163          1266  1
;  164          1267  1 ! Define entries of the allocation table.
;  165          1268  1 !
;  166          1269  1 LITERAL
;  167          1270  1         BOOTBLOCK_IDX=  0,              ! boot block
;  168          1271  1         HOMEBLOCK1_IDX= 1,              ! primary home block
;  169          1272  1         HOMEBLOCK2_IDX= 2,              ! alternate home block
;  170          1273  1         IDXHDR2_IDX=    3,              ! alternate index file header
;  171          1274  1         IDXFILE_IDX=    4,              ! index file bitmap and file headers
;  172          1275  1         BITMAP_IDX=     5,              ! storage bitmap
;  173          1276  1         MFD_IDX=        6;              ! MFD
;  174          1277  1
;  175          1278  1
;  176          1279  1 MACRO
;  177          1280  1         BOOTBLOCK_CNT=  ALLOC_TABLE_CNT[BOOTBLOCK_IDX] %,
;  178          1281  1         BOOTBLOCK_LBN=  ALLOC_TABLE_LBN[BOOTBLOCK_IDX] %,
;  179          1282  1         HOMEBLOCK1_CNT= ALLOC_TABLE_CNT[HOMEBLOCK1_IDX] %,
;  180          1283  1         HOMEBLOCK1_LBN= ALLOC_TABLE_LBN[HOMEBLOCK1_IDX] %,
;  181          1284  1         HOMEBLOCK2_CNT= ALLOC_TABLE_CNT[HOMEBLOCK2_IDX] %,
;  182          1285  1         HOMEBLOCK2_LBN= ALLOC_TABLE_LBN[HOMEBLOCK2_IDX] %,
;  183          1286  1         IDXHDR2_CNT=    ALLOC_TABLE_CNT[IDXHDR2_IDX] %,
;  184          1287  1         IDXHDR2_LBN=    ALLOC_TABLE_LBN[IDXHDR2_IDX] %,
;  185          1288  1         IDXFILE_CNT=    ALLOC_TABLE_CNT[IDXFILE_IDX] %,
;  186          1289  1         IDXFILE_LBN=    ALLOC_TABLE_LBN[IDXFILE_IDX] %,
;  187          1290  1         BITMAP_CNT=     ALLOC_TABLE_CNT[BITMAP_IDX] %,
;  188          1291  1         BITMAP_LBN=     ALLOC_TABLE_LBN[BITMAP_IDX] %,
```

```
;  189          1292  1          MFD_CNT=        ALLOC_TABLE_CNT[MFD_IDX] %,
;  190          1293  1          MFD_LBN=        ALLOC_TABLE_LBN[MFD_IDX] %;
```

```
;  192          1294  1 %SBTTL 'Boot block program'
;  193          1295  1 ! Boot program.  The following PDP-11 program will type out the attached
;  194          1296  1 ! message when the volume is booted on a PDP-11, informing the user that
;  195          1297  1 ! this is not a system disk.
;  196          1298  1 !
;  197          1299  1 BIND
;  198          1300  1         BOOT_PROGRAM    = UPLIT WORD (
;  199          1301  1
;  200          1302  1 %O'000240',                  !  BOOTBK: NOP                     ; NOP IDENTIFIES BOOT BLOCK
;  201          1303  1 %O'012706', %O'001000',      !          MOV     #1000,SP       ; SET TEMP STACK
;  202          1304  1 %O'010700',                  !          MOV     PC,R0          ; SET ADDRESS
;  203          1305  1 %O'062700', %O'000036',      !          ADD     #BOTMSG-.,R0   ; OF MESSAGE
;  204          1306  1 %O'112001',                  !  10$:    MOVB    (R0)+,R1       ; GET NEXT CHARACTER
;  205          1307  1 %O'001403',                  !          BEQ     20$            ; END
;  206          1308  1 %O'004767', %O'000006',      !          CALL    TYPIT          ; NO, PRINT IT
;  207          1309  1 %O'000773',                  !          BR      10$            ; LOOP FOR NEXT CHARACTER
;  208          1310  1 %O'000005',                  !  20$:    RESET                  ;
;  209          1311  1 %O'000000',                  !          HALT                   ; HALT
;  210          1312  1                             !
;  211          1313  1                             !
;  212          1314  1 %O'110137', %O'177566',      !  TYPIT:  MOVB    R1,@#TPB       ; PRINT CHARACTER
;  213          1315  1 %O'105737', %O'177564',      !  10$:    TSTB    @#TPS          ; DONE?
;  214          1316  1 %O'100375',                  !          BPL     10$            ; NO, WAIT
;  215          1317  1 %O'000207'                   !          RETURN                 ;
;  216          1318  1                             !
;  217          1319  1                             !
;  218          1320  1                             !  BOTMSG:
;  219          1321  1
;  220          1322  1                         );
;  221          1323  1
;  222          1324  1 LITERAL
;  223          1325  1         BOOT_PROG_LEN   = 38;
;  224          1326  1
;  225          1327  1
;  226          1328  1 ! Boot message.  Contains the volume label.
;  227          1329  1 !
;  228          1330  1 BIND
;  229          1331  1         BOOT_MESSAGE    = UPLIT BYTE (13, 10, 10,
;  230          1332  1                                       is not a system disk', 13, 10, 10, 0);
;  231          1333  1
;  232          1334  1
;  233          1335  1 LITERAL
;  234          1336  1         BOOT_MESG_LEN   = 40;
;  235          1337  1
;  236          1338  1
;  237          1339  1 MACRO
;  238          1340  1         BTB$T_VOLNAME   = 38, 0, 0, 0%; ! volume label in boot block message
;  239          1341  1 LITERAL
;  240          1342  1         BTB$S_VOLNAME   = 12;
```

```
;  242          1343  1 %SBTTL 'ALLOCATE - allocate space for table entry'
;  243          1344  1 ROUTINE ALLOCATE(INDEX): L_P$ NOVALUE=
;  244          1345  1
;  245          1346  1 !++
;  246          1347  1 !
;  247          1348  1 ! FUNCTIONAL DESCRIPTION:
;  248          1349  1 !       This routine allocates the given table entry in the first available
;  249          1350  1 !       position after the given start.  If none, it allocates in the first
;  250          1351  1 !       available position before the given start.  If none, it signals an
;  251          1352  1 !       error.
;  252          1353  1 !
;  253          1354  1 ! INPUT PAPAMETERS:
;  254          1355  1 !       INDEX           - Index of table entry to allocate.
;  255          1356  1 !
;  256          1357  1 ! IMPLICIT INPUTS:
;  257          1358  1 !       Allocation table.
;  258          1359  1 !
;  259          1360  1 ! OUTPUT PARAMETERS:
;  260          1361  1 !       NONE
;  261          1362  1 !
;  262          1363  1 ! IMPLICIT OUTPUTS:
;  263          1364  1 !       Allocation table.
;  264          1365  1 !
;  265          1366  1 ! ROUTINE VALUE:
;  266          1367  1 !       NONE
;  267          1368  1 !
;  268          1369  1 ! SIDE EFFECTS:
;  269          1370  1 !       NONE
;  270          1371  1 !
;  271          1372  1 !--
;  272          1373  1
;  273          1374  2 BEGIN
;  274          1375  2 LOCAL
;  275          1376  2         INITIAL_LBN;                    ! Initial cut at LBN
;  276          1377  2 L_DECL;
;  277          1378  2
;  278          1379  2
;  279          1380  2 ! Round the starting LBN and count down and up, respectively, to cluster
;  280          1381  2 ! boundaries.
;  281          1382  2 !
;  282          1383  2 ALLOC_TABLE_LBN[.INDEX] = .ALLOC_TABLE_LBN[.INDEX] / .CLUSTER * .CLUSTER;
;  283          1384  2 ALLOC_TABLE_CNT[.INDEX] = (.ALLOC_TABLE_CNT[.INDEX] + .CLUSTER - 1) / .CLUSTER * .CLUSTER;
;  284          1385  2 INITIAL_LBN = .ALLOC_TABLE_LBN[.INDEX];
;  285          1386  2
;  286          1387  2
;  287          1388  2 ! Iterate toward the end of the disk, checking the proposed location
;  288          1389  2 ! of the entry.
;  289          1390  2 !
;  290          1391  2 WHILE TRUE DO
;  291          1392  3     BEGIN
;  292          1393  3
;  293          1394  3     ! Try to allocate the entry.  If it succeeds, return.
;  294          1395  3     !
;  295          1396  3     IF STA_ALLOC_LBN(.ALLOC_TABLE_CNT[.INDEX], .ALLOC_TABLE_LBN[.INDEX])
;  296          1397  3     THEN
;  297          1398  3         RETURN;
;  298          1399  3
```

```
;  299          1400  3
;  300          1401  3      ! Increment to the next cluster.
;  301          1402  3      !
;  302          1403  3      ALLOC_TABLE_LBN[.INDEX] = .ALLOC_TABLE_LBN[.INDEX] + .CLUSTER;
;  303          1404  3
;  304          1405  3
;  305          1406  3      ! If we have fallen off the end of the volume, exit the loop.
;  306          1407  3      !
;  307          1408  3      IF .ALLOC_TABLE_LBN[.INDEX] GEQU .VOLUME_SIZE
;  308          1409  3      THEN
;  309          1410  3          EXITLOOP;
;  310          1411  2      END;
;  311          1412  2
;  312          1413  2
;  313          1414  2 ! Iterate toward the beginning of the disk, checking the proposed location
;  314          1415  2 ! of the entry.
;  315          1416  2 !
;  316          1417  2 ALLOC_TABLE_LBN[.INDEX] = .INITIAL_LBN;
;  317          1418  2 WHILE TRUE DO
;  318          1419  3     BEGIN
;  319          1420  3
;  320          1421  3      ! Try to allocate the entry.  If it succeeds, return.
;  321          1422  3      !
;  322          1423  3      IF STA_ALLOC_LBN(.ALLOC_TABLE_CNT[.INDEX], .ALLOC_TABLE_LBN[.INDEX])
;  323          1424  3      THEN
;  324          1425  3          RETURN;
;  325          1426  3
;  326          1427  3
;  327          1428  3      ! If we have fallen off the beginning of the volume, report failure.
;  328          1429  3      !
;  329          1430  3      IF .ALLOC_TABLE_LBN[.INDEX] EQL 0
;  330          1431  3      THEN
;  331          1432  3          SIGNAL(BACKUP$_ALLOCFAIL, 1, VCB[VCB_DEVICE]);
;  332          1433  3
;  333          1434  3
;  334          1435  3      ! Decrement to the previous cluster.
;  335          1436  3      !
;  336          1437  3      ALLOC_TABLE_LBN[.INDEX] = .ALLOC_TABLE_LBN[.INDEX] - .CLUSTER;
;  337          1438  2      END;
;  338          1439  1 END;


                                .TITLE  STAINIVOL Disk volume initialization
                                .IDENT  \V04-000\

                                .PSECT  COMMON,NOEXE,  OVR,2

                        00000 GLOBAL_BASE:
                                .BLKB   0
                        00000 FREE_LIST:
                                .BLKB   8
                        00008 INPUT_WAIT:
                                .BLKB   8
                        00010 REREAD_WAIT:
                                .BLKB   8
                        00018 OUTPUT_WAIT:
                                .BLKB   8
```

```
00020 JPI_UIC:.BLKB   4
00024 JPI_USERNAME:
                .BLKB   12
00030 JPI_DATE:
                .BLKB   8
00038 JPI_NODE_DESC:
                .BLKB   8
00040 JPI_CURPRIV:
                .BLKB   8
00048 SYI_VERSION:
                .BLKB   4
0004C SYI_SID:.BLKB   4
00050 RWSV_HOLD_LIST:
                .BLKB   8
00058 RWSV_CRC16:
                .BLKB   64
00098 RWSV_AUTODIN:
                .BLKB   64
000D8 RWSV_FILESET_ID:
                .BLKB   8
000E0 RWSV_VOLUME_ID:
                .BLKB   12
000EC RWSV_VOL_NUMBER:
                .BLKB   2
000EE RWSV_SEG_NUMBER:
                .BLKB   2
000F0 RWSV_FILE_NUMBER:
                .BLKB   4
000F4 RWSV_SAVE_QUAL:
                .BLKB   4
000F8 RWSV_SAVE_FAB:
                .BLKB   4
000FC RWSV_CHAN:
                .BLKB   4
00100 RWSV_XOR_BCB:
                .BLKB   4
00104 RWSV_IN_SEQ:
                .BLKB   4
00108 RWSV_IN_SEQ_0:
                .BLKB   4
0010C RWSV_IN_XOR_SEQ:
                .BLKB   4
00110 RWSV_IN_XOR_RFA:
                .BLKB   6
00116 RWSV_LOOKAHEAD:
                .BLKB   1
00117 RWSV_XORSIZE:
                .BLKB   1
00118 RWSV_IN_GROUP_SIZE:
                .BLKB   4
0011C RWSV_IN_ERRORS:
                .BLKB   2
0011E RWSV_IN_XORUSE:
                .BLKB   2
00120 RWSV_IN_ORGERR:
                .BLKB   8
00128 RWSV_IN_VBN:
```

```
                        .BLKB   4
0012C RWSV_IN_VBN_0:
                        .BLKB   4
00130 RWSV_ALLOC:
                        .BLKB   4
00134 RWSV_EOF:
                        .BLKB   4
00138 RWSV_OUT_SEQ:
                        .BLKB   4
0013C RWSV_OUT_VBN:
                        .BLKB   4
00140 RWSV_OUT_BLOCK_COUNT:
                        .BLKB   4
00144 RWSV_OUT_ERRORS:
                        .BLKB   2
00146 RWSV_SEQ_ERRORS:
                        .BLKB   2
00148 RWSV_OUT_GROUP_COUNT:
                        .BLKB   1
00149 RWSV_PADDING:
                        .BLKB   3
0014C QUAL:             .BLKB   112
001BC COM_SSNAME:
                        .BLKB   8
001C4 COM_VALID_TYPES:
                        .BLKB   2
001C6 COM_FLAGS:
                        .BLKB   2
001C8 COM_PADDING:
                        .BLKB   1
001C9 COM_BUFF_COUNT:
                        .BLKB   1
001CA COM_I_SETCOUNT:
                        .BLKB   1
001CB COM_O_SETCOUNT:
                        .BLKB   1
001CC COM_I_STRUCNAME:
                        .BLKB   12
001D8 COM_O_STRUCNAME:
                        .BLKB   12
001E4 COM_O_BSRDATE:
                        .BLKB   8
001EC ALT_SSNAME:
                        .BLKB   32
0020C INPUT_FUNC:
                        .BLKB   1
0020D INPUT_RTYPE:
                        .BLKB   1
0020E OUTPUT_FUNC:
                        .BLKB   1
0020F FAST_STRUCLEV:
                        .BLKB   1
00210 INPUT_BEG:
                        .BLKB   0
00210 INPUT_CHAN:
                        .BLKB   4
00214 INPUT_FLAGS:
```

```
                        .BLKB   2
00216 INPUT_PADDING:
                        .BLKB   2
00218 INPUT_FAB:
                        .BLKB   4
0021C INPUT_NAM:
                        .BLKB   4
00220 INPUT_BCB:
                        .BLKB   4
00224 INPUT_QUAL:
                        .BLKB   4
00228 INPUT_BAD:
                        .BLKB   4
0022C INPUT_BLOCK:
                        .BLKB   4
00230 INPUT_MAXBLOCK:
                        .BLKB   4
00234 INPUT_MEDIA_ID:
                        .BLKB   4
00238 INPUT_NAMEDESC:
                        .BLKB   8
00240 INPUT_STATBLK:
                        .BLKB   8
00248 INPUT_HDR_BEG:
                        .BLKB   0
00248 INPUT_CREDATE:
                        .BLKB   8
00250 INPUT_REVDATE:
                        .BLKB   8
00258 INPUT_EXPDATE:
                        .BLKB   8
00260 INPUT_BAKDATE:
                        .BLKB   8
00268 INPUT_FILEOWNER:
                        .BLKB   4
0026C INPUT_FILECHAR:
                        .BLKB   4
00270 INPUT_RECATTR:
                        .BLKB   32
00290 INPUT_HDR_END:
                        .BLKB   0
00290 INPUT_END:
                        .BLKB   0
00290 INPUT_PROC_LIST:
                        .BLKB   4
00294 INPUT_PLACEMENT:
                        .BLKB   8
0029C INPUT_VBN_LIST:
                        .BLKB   8
002A4 INPUT_PLACE_LEN:
                        .BLKB   2
002A6 INPUT_PADDING_2:
                        .BLKB   2
002A8 OUTPUT_BEG:
                        .BLKB   0
002A8 OUTPUT_CHAN:
                        .BLKB   4
```

```
002AC OUTPUT_FLAGS:
                .BLKB   2
002AE OUTPUT_PADDING:
                .BLKB   2
002B0 OUTPUT_FAB:
                .BLKB   4
002B4 OUTPUT_NAM:
                .BLKB   4
002B8 OUTPUT_BCB:
                .BLKB   4
002BC OUTPUT_QUAL:
                .BLKB   4
002C0 OUTPUT_BAD:
                .BLKB   4
002C4 OUTPUT_BLOCK:
                .BLKB   4
002C8 OUTPUT_MAXBLOCK:
                .BLKB   4
002CC OUTPUT_DEVGEOM:
                .BLKB   8
002D4 OUTPUT_ATTBUF:
                .BLKB   144
00364 OUTPUT_END:
                .BLKB   0
00364 LIST_TOTFILES:
                .BLKB   4
00368 LIST_TOTSIZE:
                .BLKB   4
0036C VERIFY_FAB:
                .BLKB   4
00370 VERIFY_USE_COUNT:
                .BLKB   4
00374 VERIFY_QUAL:
                .BLKB   4
00378 COMPARE_BCB:
                .BLKB   4
0037C FAST_BUFFER:
                .BLKB   4
00380 FAST_BUFFER_SIZE:
                .BLKB   4
00384 FAST_RVN:
                .BLKB   1
00385 FAST_PADDING:
                .BLKB   1
00386 DIR_VERLIMIT:
                .BLKB   2
00388 FAST_VOL_BEG:
                .BLKB   0
00388 FAST_IMAP_SIZE:
                .BLKB   4
0038C FAST_IMAP:
                .BLKB   4
00390 FAST_HDR_OFFSET:
                .BLKB   4
00394 FAST_BOOT_LBN:
                .BLKB   4
00398 FAST_VOL_END:
```

```
                        .BLKB   0
00398 JOUR_BUFFER:
                        .BLKB   4
0039C JOUR_DIR:
                        .BLKB   4
003A0 JOUR_HIBLK:
                        .BLKB   4
003A4 JOUR_EFBLK:
                        .BLKB   4
003A8 JOUR_INBLK:
                        .BLKB   4
003AC JOUR_FFBYTE:
                        .BLKB   2
003AE JOUR_INBYTE:
                        .BLKB   2
003B0 JOUR_STRUCT_LEV:
                        .BLKB   2
003B2 JOUR_COUNT:
                        .BLKB   1
003B3 JOUR_REVERSE:
                        .BLKB   1
003B4 JOUR_EXSZ:
                        .BLKB   2
003B6 JOUR_PADDING:
                        .BLKB   2
003B8 CHKPT_HIGH_SP:
                        .BLKB   4
003BC CHKPT_LOW_SP:
                        .BLKB   4
003C0 CHKPT_STACK:
                        .BLKB   4
003C4 CHKPT_VARS:
                        .BLKB   4
003C8 CHKPT_STATUS:
                        .BLKB   4
003CC DIR_BEG:          .BLKB   0
003CC DIR_CHAN:
                        .BLKB   4
003D0 DIR_NAM:          .BLKB   4
003D4 DIR_DEV_DESC:
                        .BLKB   4
003D8 DIR_SEL_DIR:
                        .BLKB   8
003E0 DIR_SEL_NTV:
                        .BLKB   8
003E8 DIR_STRUCLEV:
                        .BLKB   1
003E9 DIR_LEVELS:
                        .BLKB   1
003EA DIR_FLAGS:
                        .BLKB   1
003EB DIR_STATUS:
                        .BLKB   1
003EC DIR_STRING:
                        .BLKB   320
0052C DIR_STACK:
                        .BLKB   612
```

```
00790 DIR_SP: .BLKB   4
00794 DIR_SEL_LATEST:
                .BLKB   4
00798 DIR_END:.BLKB   0
00798 DIR_SCANLIMIT:
                .BLKB   36
007BC INPUT_MTL:
                .BLKB   4
007C0 OUTPUT_MTL:
                .BLKB   4
007C4 CURRENT_MTL:
                .BLKB   4
007C8 CURRENT_VCB:
                .BLKB   4
007CC CURRENT_WCB:
                .BLKB   4
007D0 ACL_FIB_DESCR:
                .BLKB   8
007D8 ACL_FIB:.BLKB   64
00818 ACL_LENGTH:
                .BLKB   4
0081C ACL_BUFFER:
                .BLKB   4
00820 CRYP_IN_CONTEXT:
                .BLKB   4
00824 CRYP_OU_CONTEXT:
                .BLKB   4
00828 CRYP_DA_CONTEXT:
                .BLKB   4
0082C CRYP_DATA_ENCIV:
                .BLKB   8
00834 CRYP_DATA_CODE:
                .BLKB   4
00838 CRYP_DATA_KEY:
                .BLKB   8
00840 CRYP_DATA_IV:
                .BLKB   8
00848 CRYP_DATA_CKSM:
                .BLKB   4

                .PSECT  CODE,NOWRT,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 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

                                                BOOT_PROGRAM=       P.AAA
                                                BOOT_MESSAGE=       P.AAB
                                                .EXTRN  FREE_VM, GET_VM
                                                .EXTRN  GET_ZERO_VM, CHECKSUM
                                                .EXTRN  CHECKSUM2, CREATE_WINDOW
                                                .EXTRN  GET_BADBLOCKS, MAKE_POINTER1
```

```
                                                        .EXTRN  MAKE_POINTER, STA_ALLOC_LBN
                                                        .EXTRN  TO_ODS1_DATE, BACKUP$_ALLOCFAIL
                                                        .EXTRN  BACKUP$_BLKZERO
                                                        .EXTRN  BACKUP$_CLUSTER
                                                        .EXTRN  BACKUP$_LARGECNT
                                                        .EXTRN  BACKUP$_MAXBAD, BACKUP$_WRITEERR

                                        007C 00000 ALLOCATE:
                                                        .WORD   Save R2,R3,R4,R5,R6                     ; 1344
                    56 00000000G   00    9E 00002           MOVAB   STA_ALLOC_LBN, R6                  ;
                    50       04    AC    D0 00009           MOVL    INDEX, R0                          ; 1383
                    52       3C  AB40    DE 0000D           MOVAL   60(P$)[R0], R2                     ;
                    53       14    AB    D0 00012           MOVL    20(P$), R3                         ;
              51    62              53    C7 00016           DIVL3   R3, (R2), R1                       ;
              62    51              53    C5 0001A           MULL3   R3, R1, (R2)                       ;
                    54       20  AB40    DE 0001E           MOVAL   32(P$)[R0], R4                     ; 1384
              50    64              53    C1 00023           ADDL3   R3, (R4), R0                       ;
                                    50    D7 00027           DECL    R0                                 ;
                    50              53    C6 00029           DIVL2   R3, R0                             ;
              64    50              53    C5 0002C           MULL3   R3, R0, (R4)                       ;
                    55              62    D0 00030           MOVL    (R2), INITIAL_LBN                  ; 1385
                                    62    DD 00033 1$:       PUSHL   (R2)                               ; 1396
                                    64    DD 00035           PUSHL   (R4)                               ;
                    66              02    FB 00037           CALLS   #2, STA_ALLOC_LBN                  ;
                    33              50    E8 0003A           BLBS    R0, 4$                             ;
                    62              53    C0 0003D           ADDL2   R3, (R2)                           ; 1403
              0C    AB              62    D1 00040           CMPL    (R2), 12(P$)                       ; 1408
                                    ED    1F 00044           BLSSU   1$                                 ;
                    62              55    D0 00046           MOVL    INITIAL_LBN, (R2)                  ; 1417
                                    62    DD 00049 2$:       PUSHL   (R2)                               ; 1423
                                    64    DD 0004B           PUSHL   (R4)                               ;
                    66              02    FB 0004D           CALLS   #2, STA_ALLOC_LBN                  ;
                    1D              50    E8 00050           BLBS    R0, 4$                             ;
                                    62    D5 00053           TSTL    (R2)                               ; 1430
                                    14    12 00055           BNEQ    3$                                 ;
              7E          18    AB        20 C1 00057        ADDL3   #32, 24(P$), -(SP)                 ; 1432
                                          01 DD 0005C        PUSHL   #1                                 ;
                                00000000G 8F DD 0005E        PUSHL   #BACKUP$_ALLOCFAIL                 ;
                00000000G       00        03 FB 00064        CALLS   #3, LIB$SIGNAL                     ;
                                62        53 C2 0006B 3$:    SUBL2   R3, (R2)                           ; 1437
                                          D9 11 0006E        BRB     2$                                 ; 1418
                                             04 00070 4$:    RET                                        ; 1439

; Routine Size:  113 bytes,    Routine Base:  CODE + 004E
```

```
;  340          1440  1 %SBTTL 'ALLOCATE_HOME - allocate home block'
;  341          1441  1 ROUTINE ALLOCATE_HOME(INDEX): L_P$ NOVALUE=
;  342          1442  1
;  343          1443  1 !++
;  344          1444  1 !
;  345          1445  1 ! FUNCTIONAL DESCRIPTION:
;  346          1446  1 !       This routine allocates the indicated allocation table entry to
;  347          1447  1 !       the first available block on the home block search sequence.
;  348          1448  1 !
;  349          1449  1 ! INPUT PARAMETERS:
;  350          1450  1 !       INDEX           - Table index of home block cluster.
;  351          1451  1 !
;  352          1452  1 ! IMPLICIT INPUTS:
;  353          1453  1 !       Allocation table.
;  354          1454  1 !
;  355          1455  1 ! OUTPUT PARAMETERS:
;  356          1456  1 !       NONE
;  357          1457  1 !
;  358          1458  1 ! IMPLICIT OUTPUTS:
;  359          1459  1 !       Allocation table.
;  360          1460  1 !
;  361          1461  1 ! ROUTINE VALUE:
;  362          1462  1 !       NONE
;  363          1463  1 !
;  364          1464  1 ! SIDE EFFECTS:
;  365          1465  1 !       NONE
;  366          1466  1 !
;  367          1467  1 !--
;  368          1468  1
;  369          1469  2 BEGIN
;  370          1470  2 LOCAL
;  371          1471  2         DELTA,                          ! home block search delta
;  372          1472  2         BLOCKFACT,                      ! device blocking factor
;  373          1473  2         LBN;                            ! home block candidate LBN
;  374          1474  2 L_DECL;
;  375          1475  2
;  376          1476  2
;  377          1477  2 ! Compute the home block search delta.  For structure level 1, this is simply
;  378          1478  2 ! 256, except that the first slot is on LBN 1 rather than 0.  For level 2,
;  379          1479  2 ! compute the home block search delta from the volume geometry, according to
;  380          1480  2 ! the following rules, where volume geometry is expressed in the order
;  381          1481  2 ! sectors, tracks, cylinders:
;  382          1482  2 !
;  383          1483  2 !       n x 1 x 1:      1
;  384          1484  2 !       1 x n x 1:      1
;  385          1485  2 !       1 x 1 x n:      1
;  386          1486  2 !
;  387          1487  2 !       n x m x 1:      n+1
;  388          1488  2 !       n x 1 x m:      n+1
;  389          1489  2 !       1 x n x m:      n+1
;  390          1490  2 !
;  391          1491  2 !       s x t x c:      (t+1)*s+1
;  392          1492  2 !
;  393          1493  2 IF .STRUCLEV_1
;  394          1494  2 THEN
;  395          1495  2     DELTA = 256
;  396          1496  2 ELSE
```

```
;  397          1497  3     BEGIN
;  398          1498  4     BLOCKFACT = (.DEVCHAR[DIB$B_SECTORS]
;  399          1499  4                 * .DEVCHAR[DIB$B_TRACKS]
;  400          1500  4                 * .DEVCHAR[DIB$W_CYLINDERS])
;  401          1501  3                 / .DEVCHAR[DIB$L_MAXBLOCK];
;  402          1502  3     DELTA = 1;
;  403          1503  3     IF
;  404          1504  3         .DEVCHAR[DIB$W_CYLINDERS] GTR 1 AND
;  405          1505  3         .DEVCHAR[DIB$B_TRACKS] GTR 1
;  406          1506  3     THEN
;  407          1507  3         DELTA = .DELTA * .DEVCHAR[DIB$B_TRACKS];
;  408          1508  3     IF
;  409          1509  3         .DEVCHAR[DIB$B_SECTORS] GTR 1 AND
;  410          1510  4         (.DEVCHAR[DIB$W_CYLINDERS] GTR 1 OR
;  411          1511  4             .DEVCHAR[DIB$B_TRACKS] GTR 1)
;  412          1512  3     THEN
;  413          1513  3         DELTA = (.DELTA * .DEVCHAR[DIB$B_SECTORS] + .BLOCKFACT) / .BLOCKFACT;
;  414          1514  3     IF
;  415          1515  3         .DELTA EQL 0 OR
;  416          1516  3         .DELTA GTRU .DEVCHAR[DIB$L_MAXBLOCK] / 10
;  417          1517  3     THEN
;  418          1518  3         DELTA = 1;
;  419          1519  2     END;
;  420          1520  2 HOMEBLOCK_DELTA = .DELTA;
;  421          1521  2
;  422          1522  2
;  423          1523  2 ! Now find the first available cluster on the search sequence by starting
;  424          1524  2 ! with LBN 1 and incrementing by the delta for each try.  If none is available,
;  425          1525  2 ! report failure.
;  426          1526  2 !
;  427          1527  2 LBN = 1;
;  428          1528  2 WHILE TRUE DO
;  429          1529  3     BEGIN
;  430          1530  3     ALLOC_TABLE_LBN[.INDEX] = .LBN / .CLUSTER * .CLUSTER;
;  431          1531  3     IF STA_ALLOC_LBN(.CLUSTER, .ALLOC_TABLE_LBN[.INDEX]) THEN EXITLOOP;
;  432          1532  3     IF .STRUCLEV_1 THEN LBN = .LBN AND NOT 7;
;  433          1533  3     LBN = .LBN + .DELTA;
;  434          1534  3     IF .LBN GEQU .VOLUME_SIZE
;  435          1535  3     THEN
;  436          1536  3         SIGNAL(BACKUP$_ALLOCFAIL, 1, VCB[VCB_DEVICE]);
;  437          1537  2     END;
;  438          1538  2
;  439          1539  2
;  440          1540  2 ! Save the LBN of the actual block.
;  441          1541  2 !
;  442          1542  2 REAL_HOMEBLOCK = .LBN;
;  443          1543  1 END;


                                    001C 00000 ALLOCATE_HOME:
                                                       .WORD   Save R2,R3,R4                          ; 1441
                                07       04    AB  E9 00002                BLBC    4(P$), 1$              ; 1493
                                53     0100    8F  3C 00006                MOVZWL  #256, DELTA            ; 1495
                                               63  11 0000B                BRB     6$                     ;
```

```
                                50      1C   AB  D0 0000D 1$:     MOVL    28(P$), R0                              ; 1498
                                51      08   A0  9A 00011        MOVZBL  8(R0), R1                               ; 1499
                                52      09   A0  9A 00015        MOVZBL  9(R0), R2                               ;
                                51           52  C4 00019        MULL2   R2, R1                                  ;
                                54      0A   A0  3C 0001C        MOVZWL  10(R0), R4                              ; 1500
                                51           54  C4 00020        MULL2   R4, R1                                  ;
                          52    51      70   A0  C7 00023        DIVL3   112(R0), R1, BLOCKFACT                  ; 1501
                                53           01  D0 00028        MOVL    #1, DELTA                               ; 1502
                                             51  D4 0002B        CLRL    R1                                      ; 1504
                                01      0A   A0  B1 0002D        CMPW    10(R0), #1                              ;
                                             0F  1B 00031        BLEQU   2$                                      ;
                                             51  D6 00033        INCL    R1                                      ;
                                01      09   A0  91 00035        CMPB    9(R0), #1                               ; 1505
                                             07  1B 00039        BLEQU   2$                                      ;
                                54      09   A0  9A 0003B        MOVZBL  9(R0), R4                               ; 1507
                                53           54  C0 0003F        ADDL2   R4, DELTA                               ;
                                01      08   A0  91 00042 2$:     CMPB    8(R0), #1                               ; 1509
                                             17  1B 00046        BLEQU   4$                                      ;
                                06           51  E8 00048        BLBS    R1, 3$                                  ; 1510
                                01      09   A0  91 0004B        CMPB    9(R0), #1                               ; 1511
                                             0E  1B 0004F        BLEQU   4$                                      ;
                                51      08   A0  9A 00051 3$:     MOVZBL  8(R0), R1                               ; 1513
                                51           53  C4 00055        MULL2   DELTA, R1                               ;
                                51           52  C0 00058        ADDL2   BLOCKFACT, R1                           ;
                          53    51           52  C7 0005B        DIVL3   BLOCKFACT, R1, DELTA                    ;
                                             53  D5 0005F 4$:     TSTL    DELTA                                   ; 1515
                                             0A  13 00061        BEQL    5$                                      ;
                          50    70   A0            0A  C7 00063        DIVL3   #10, 112(R0), R0                        ; 1516
                                50           53  D1 00068        CMPL    DELTA, R0                               ;
                                             03  1B 0006B        BLEQU   6$                                      ;
                                53           01  D0 0006D 5$:     MOVL    #1, DELTA                               ; 1518
                                      08     AB  53  D0 00070 6$:     MOVL    DELTA, 8(P$)                            ; 1520
                                52           01  D0 00074        MOVL    #1, LBN                                 ; 1527
                                54      04   AC  D0 00077        MOVL    INDEX, R4                               ; 1530
                          50    52      14   AB  C7 0007B 7$:     DIVL3   20(P$), LBN, R0                         ;
                    3C AB44     50      14   AB  C5 00080        MULL3   20(P$), R0, 60(P$)[R4]                  ;
                                         3C AB44 DD 00087        PUSHL   60(P$)[R4]                              ; 1531
                                         14   AB  DD 0008B        PUSHL   20(P$)                                  ;
                          00000000G 00           02  FB 0008E        CALLS   #2, STA_ALLOC_LBN                       ;
                                50           50  E8 00095        BLBS    R0, 9$                                  ;
                                03      04   AB  E9 00098        BLBC    4(P$), 8$                               ; 1532
                                52           01  8A 0009C        BICB2   #1, LBN                                 ;
                                52           53  C0 0009F 8$:     ADDL2   DELTA, LBN                              ; 1533
                                      0C     AB  52  D1 000A2        CMPL    LBN, 12(P$)                             ; 1534
                                             D3  1F 000A6        BLSSU   7$                                      ;
                          7E    18     AB            20  C1 000A8        ADDL3   #32, 24(P$), -(SP)                      ; 1536
                                             01  DD 000AD        PUSHL   #1                                      ;
                                00000000G    8F  DD 000AF        PUSHL   #BACKUP$_ALLOCFAIL                      ;
                          00000000G 00           03  FB 000B5        CALLS   #3, LIB$SIGNAL                          ;
                                             BD  11 000BC        BRB     7$                                      ; 1528
                                      10     AB  52  D0 000BE 9$:     MOVL    LBN, 16(P$)                             ; 1542
                                                 04  000C2        RET                                             ; 1543

; Routine Size:  195 bytes,    Routine Base:  CODE + 00BF
```

```
;  445          1544  1 %SBTTL 'INIT_ALLOCATE - allocate space for volume structures'
;  446          1545  1 ROUTINE INIT_ALLOCATE: L_P$ NOVALUE=
;  447          1546  1
;  448          1547  1 !++
;  449          1548  1 !
;  450          1549  1 ! FUNCTIONAL DESCRIPTION:
;  451          1550  1 !       This is the main allocation routine.  It determines the size and
;  452          1551  1 !       location of each portion of the file structure.  Each allocation is
;  453          1552  1 !       done by choosing a candidate location for the section and checking
;  454          1553  1 !       for conflicts.  If a conflict exists, a new location as near as
;  455          1554  1 !       possible to the desired one is chosen.
;  456          1555  1 !
;  457          1556  1 ! INPUT PARAMETERS:
;  458          1557  1 !       NONE
;  459          1558  1 !
;  460          1559  1 ! IMPLICIT INPUTS:
;  461          1560  1 !       NONE
;  462          1561  1 !
;  463          1562  1 ! OUTPUT PARAMETERS:
;  464          1563  1 !       NONE
;  465          1564  1 !
;  466          1565  1 ! IMPLICIT OUTPUTS:
;  467          1566  1 !       NONE
;  468          1567  1 !
;  469          1568  1 ! ROUTINE VALUE:
;  470          1569  1 !       NONE
;  471          1570  1 !
;  472          1571  1 ! SIDE EFFECTS:
;  473          1572  1 !       NONE
;  474          1573  1 !
;  475          1574  1 !--
;  476          1575  1
;  477          1576  2 BEGIN
;  478          1577  2 LOCAL
;  479          1578  2         BAD:            REF BBLOCK;     ! Pointer to bad block descriptor
;  480          1579  2 L_DECL;
;  481          1580  2
;  482          1581  2
;  483          1582  2 ! Allocate the bad blocks to the bad block file.
;  484          1583  2 !
;  485          1584  2 BAD = OUTPUT_BAD[BAD_DESC];
;  486          1585  2 INCR J FROM 0 TO .OUTPUT_BAD[BAD_NUMDESC]-1 DO
;  487          1586  3     BEGIN
;  488          1587  3     IF NOT STA_ALLOC_LBN(.BAD[BAD_COUNT], .BAD[BAD_LBN])
;  489          1588  3     THEN
;  490          1589  3         SIGNAL(BACKUP$_ALLOCFAIL, 1, VCB[VCB_DEVICE]);
;  491          1590  3     BAD = .BAD + BAD_S_DESC;
;  492          1591  2     END;
;  493          1592  2
;  494          1593  2
;  495          1594  2 ! Allocate the boot block to the first available cluster (usually 0).
;  496          1595  2 !
;  497          1596  2 BOOTBLOCK_CNT = .CLUSTER;
;  498          1597  2 ALLOCATE(BOOTBLOCK_IDX);
;  499          1598  2 IF .BOOTBLOCK_LBN NEQ 0
;  500          1599  2 THEN
;  501          1600  2     SIGNAL(BACKUP$_BLKZERO, 1, VCB[VCB_DEVICE]);
```

```
;  502          1601  2
;  503          1602  2
;  504          1603  2 ! Next allocate the primary and secondary home blocks.  If the boot block is
;  505          1604  2 ! on LBN 0 and the cluster factor is greater than 1, then the primary home
;  506          1605  2 ! block cluster is a dummy since the real home block is LBN 1.
;  507          1606  2 !
;  508          1607  2 HOMEBLOCK1_CNT = .CLUSTER;
;  509          1608  2 IF .STRUCLEV_1
;  510          1609  2 THEN
;  511          1610  2     ALLOCATE_HOME(HOMEBLOCK1_IDX)
;  512          1611  2 ELSE
;  513          1612  3     BEGIN
;  514          1613  3     IF .BOOTBLOCK_LBN EQL 0 AND .CLUSTER GTR 1
;  515          1614  3     THEN
;  516          1615  3         ALLOCATE(HOMEBLOCK1_IDX)
;  517          1616  3     ELSE
;  518          1617  3         ALLOCATE_HOME(HOMEBLOCK1_IDX);
;  519          1618  3     HOMEBLOCK2_CNT = .CLUSTER;
;  520          1619  3     ALLOCATE_HOME(HOMEBLOCK2_IDX);
;  521          1620  2     END;
;  522          1621  2
;  523          1622  2
;  524          1623  2 ! Now allocate the MFD, storage map, index file, and alternate index file
;  525          1624  2 ! header, in that order.  This results in optimal locality of the most
;  526          1625  2 ! frequently referenced portions of the file structure.  We do not allocate
;  527          1626  2 ! the MFD at this time if this is an output disk being restored, since it
;  528          1627  2 ! is allocated when it is encountered in the save set.
;  529          1628  2 !
;  530          1629  2 IF .VCB[VCB_SAVESET]
;  531          1630  2 THEN
;  532          1631  3     BEGIN
;  533          1632  3     MFD_LBN = .OUTPUT_ATTBUF[VSR_INDEXLBN];
;  534          1633  3     MFD_CNT = 1;
;  535          1634  2     END;
;  536          1635  2
;  537          1636  2 ALLOCATE(MFD_IDX);
;  538          1637  2 BITMAP_LBN = .OUTPUT_ATTBUF[VSR_INDEXLBN];
;  539          1638  2 BITMAP_CNT = ((.VOLUME_SIZE/.CLUSTER + 4095) / 4096) + 1;
;  540          1639  2 ALLOCATE(BITMAP_IDX);
;  541          1640  2 VCB[VCB_BITMAP_LBN] = .BITMAP_LBN + 1;
;  542          1641  2 VCB[VCB_BITMAP_SIZE] = .BITMAP_CNT - 1;
;  543          1642  2
;  544          1643  2
;  545          1644  2 IDXFILE_LBN = .OUTPUT_ATTBUF[VSR_INDEXLBN];
;  546          1645  2 IDXFILE_CNT = .OUTPUT_ATTBUF[VSR_MAXFILNUM] + (.OUTPUT_ATTBUF[VSR_MAXFILES]+4095)/4096;
;  547          1646  2 ALLOCATE(IDXFILE_IDX);
;  548          1647  2
;  549          1648  2
;  550          1649  2 IF NOT .STRUCLEV_1
;  551          1650  2 THEN
;  552          1651  3     BEGIN
;  553          1652  3     IDXHDR2_CNT = .CLUSTER;
;  554          1653  3     IDXHDR2_LBN = .IDXFILE_LBN + .HOMEBLOCK_DELTA;
;  555          1654  3     ALLOCATE(IDXHDR2_IDX);
;  556          1655  2     END;
;  557          1656  1 END;
```

```
                                00FC 00000 INIT_ALLOCATE:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1545
                57 00000000G    00   9E 00002          MOVAB   LIB$SIGNAL, R7              ;
                56     FEBF     CF   9E 00009          MOVAB   ALLOCATE, R6                ;
                55 00000000'    EF   9E 0000E          MOVAB   OUTPUT_ATTBUF+84, R5        ;
52        98    A5              08   C1 00015          ADDL3   #8, OUTPUT_BAD, BAD         ; 1584
                54        98    B5   D0 0001A          MOVL    @OUTPUT_BAD, R4             ; 1585
                53              01   CE 0001E          MNEGL   #1, J                       ;
                                22   11 00021          BRB     3$                          ;
                                62   DD 00023 1$:      PUSHL   (BAD)                       ; 1587
                        04      A2   DD 00025          PUSHL   4(BAD)                      ;
   00000000G    00              02   FB 00028          CALLS   #2, STA_ALLOC_LBN           ;
                10              50   E8 0002F          BLBS    R0, 2$                      ;
7E        18    AB              20   C1 00032          ADDL3   #32, 24(P$), -(SP)          ; 1589
                                01   DD 00037          PUSHL   #1                          ;
                00000000G       8F   DD 00039          PUSHL   #BACKUP$_ALLOCFAIL          ;
                67              03   FB 0003F          CALLS   #3, LIB$SIGNAL              ;
                52              08   C0 00042 2$:      ADDL2   #8, BAD                     ; 1590
DA              53              54   F2 00045 3$:      AOBLSS  R4, J, 1$                   ; 1585
          20    AB        14    AB   D0 00049          MOVL    20(P$), 32(P$)              ; 1596
                                7E   D4 0004E          CLRL    -(SP)                       ; 1597
                66              01   FB 00050          CALLS   #1, ALLOCATE                ;
                        3C      AB   D5 00053          TSTL    60(P$)                      ; 1598
                                10   13 00056          BEQL    4$                          ;
7E        18    AB              20   C1 00058          ADDL3   #32, 24(P$), -(SP)          ; 1600
                                01   DD 0005D          PUSHL   #1                          ;
                00000000G       8F   DD 0005F          PUSHL   #BACKUP$_BLKZERO            ;
                67              03   FB 00065          CALLS   #3, LIB$SIGNAL              ;
          24    AB        14    AB   D0 00068 4$:      MOVL    20(P$), 36(P$)              ; 1607
          04    AB        04    AB   E9 0006D          BLBC    4(P$), 5$                   ; 1608
                                01   DD 00071          PUSHL   #1                          ; 1610
                                1F   11 00073          BRB     8$                          ;
                        3C      AB   D5 00075 5$:      TSTL    60(P$)                      ; 1613
                                0D   12 00078          BNEQ    6$                          ;
                01      14      AB   D1 0007A          CMPL    20(P$), #1                  ;
                                07   15 0007E          BLEQ    6$                          ;
                                01   DD 00080          PUSHL   #1                          ; 1615
                66              01   FB 00082          CALLS   #1, ALLOCATE                ;
                                06   11 00085          BRB     7$                          ;
                                01   DD 00087 6$:      PUSHL   #1                          ; 1617
          71    A6              01   FB 00089          CALLS   #1, ALLOCATE_HOME           ;
          28    AB        14    AB   D0 0008D 7$:      MOVL    20(P$), 40(P$)              ; 1618
                                02   DD 00092          PUSHL   #2                          ; 1619
          71    A6              01   FB 00094 8$:      CALLS   #1, ALLOCATE_HOME           ;
                50        18    AB   D0 00098          MOVL    24(P$), R0                  ; 1629
08        07    A0              03   E1 0009C          BBC     #3, 7(R0), 9$               ;
          54    AB              65   D0 000A1          MOVL    OUTPUT_ATTBUF+84, 84(P$)    ; 1632
          38    AB              01   D0 000A5          MOVL    #1, 56(P$)                  ; 1633
                                06   DD 000A9 9$:      PUSHL   #6                          ; 1636
                66              01   FB 000AB          CALLS   #1, ALLOCATE                ;
          50    AB              65   D0 000AE          MOVL    OUTPUT_ATTBUF+84, 80(P$)    ; 1637
50        0C    AB        14    AB   C7 000B2          DIVL3   20(P$), 12(P$), R0          ; 1638
                50      0FFF    C0   9E 000B8          MOVAB   4095(R0), R0                ;
```

```
                                50 00001000    8F  C6 000BD          DIVL2   #4096, R0                                          ;
                             34 AB       01    A0  9E 000C4          MOVAB   1(R0), 52(P$)                                      ;
                                             05  DD 000C9          PUSHL   #5                                                 ; 1639
                                66           01  FB 000CB          CALLS   #1, ALLOCATE                                       ;
                                50       18  AB  D0 000CE          MOVL    24(P$), R0                                         ; 1640
                          0C    A0     50  AB      01  C1 000D2    ADDL3   #1, 80(P$), 12(R0)                                 ;
                          34    A0     34  AB      01  A3 000D8    SUBW3   #1, 52(P$), 52(R0)                                 ; 1641
                                4C  AB          65  D0 000DE      MOVL    OUTPUT_ATTBUF+84, 76(P$)                           ; 1644
                                50     E0  A5 00000FFF    8F  C1 000E2    ADDL3   #4095, OUTPUT_ATTBUF+52, R0                ; 1645
                                50 00001000    8F  C6 000EB    DIVL2   #4096, R0                                          ;
                             30 AB       E4 B540    9E  00F2    MOVAB   @OUTPUT_ATTBUF+56[R0], 48(P$)                      ;
                                             04  DD 000F8          PUSHL   #4                                                 ; 1646
                                66           01  FB 000FA          CALLS   #1, ALLOCATE                                       ;
                                11       04  AB  E8 000FD          BLBS    4(P$), 10$                                         ; 1649
                             2C AB       14  AB  D0 00101          MOVL    20(P$), 44(P$)                                     ; 1652
                          48    AB     4C  AB      08  AB  C1 00106    ADDL3   8(P$), 76(P$), 72(P$)                          ; 1653
                                             03  DD 0010D          PUSHL   #3                                                 ; 1654
                                66           01  FB 0010F          CALLS   #1, ALLOCATE                                       ;
                                             04 00112 10$:      RET                                                        ; 1656

; Routine Size:  275 bytes,    Routine Base:  CODE + 0182
```

```
;  559          1657  1 %SBTTL 'WRITE_BLOCK - write block to volume'
;  560          1658  1 ROUTINE WRITE_BLOCK(LBN,BUFFER): L_PS NOVALUE=
;  561          1659  1
;  562          1660  1 !++
;  563          1661  1 !
;  564          1662  1 ! FUNCTIONAL DESCRIPTION:
;  565          1663  1 !       This routine writes a disk block by logical block number.
;  566          1664  1 !
;  567          1665  1 ! INPUT PARAMETERS:
;  568          1666  1 !       LBN             - Logical block number.
;  569          1667  1 !       BUFFER          - Poirter to buffer.
;  570          1668  1 !
;  571          1669  1 ! IMPLICIT INPUTS:
;  572          1670  1 !       NONE
;  573          1671  1 !
;  574          1672  1 ! OUTPUT PARAMETERS:
;  575          1673  1 !       NONE
;  576          1674  1 !
;  577          1675  1 ! IMPLICIT OUTPUTS:
;  578          1676  1 !       NONE
;  579          1677  1 !
;  580          1678  1 ! ROUTINE VALUE:
;  581          1679  1 !       NONE
;  582          1680  1 !
;  583          1681  1 ! SIDE EFFECTS:
;  584          1682  1 !       NONE
;  585          1683  1 !
;  586          1684  1 !--
;  587          1685  1
;  588          1686  2 BEGIN
;  589          1687  2 LOCAL
;  590          1688  2         STATUS,                         ! System service status
;  591          1689  2         IOSB:           VECTOR[4,WORD]; ! I/O status block
;  592          1690  2 L_DECL;
;  593          1691  2
;  594          1692  2
;  595        P 1693  2 STATUS = $QIOW(
;  596        P 1694  2     FUNC=IO$_WRITELBLK,
;  597        P 1695  2     CHAN=.CHANNEL,
;  598        P 1696  2     IOSB=IOSB,
;  599        P 1697  2     P1=.BUFFER,
;  600        P 1698  2     P2=512,
;  601          1699  2     P3=.LBN);
;  602          1700  2 IF .STATUS THEN STATUS = .IOSB[0];
;  603          1701  2 IF NOT .STATUS
;  604          1702  2 THEN
;  605          1703  2     SIGNAL(BACKUP$_WRITEERR + STS$K_SEVERE, 1, VCB[VCB_DEVICE], .STATUS);
;  606          1704  1 END;


                                                .EXTRN  SYS$QIOW

                                0000 00000 WRITE_BLOCK:
                                                .WORD   Save nothing                                    ; 1658
                             5E                 08  C2 00002          SUBL2   #8, SP                  ;
                                                7E  7C 00005          CLRQ    -(SP)                   ; 1699
```

```
                                                7E  D4 00007          CLRL    -(SP)                                          ;
                                             04 AC  DD 00009          PUSHL   LBN                                            ;
                                   7E      0200 8F  3C 0000C          MOVZWL  #512, -(SP)                                    ;
                                             08 AC  DD 00011          PUSHL   BUFFER                                         ;
                                                7E  7C 00014          CLRQ    -(SP)                                          ;
                                             20 AE  9F 00016          PUSHAB  IOSB                                           ;
                                                20  DD 00019          PUSHL   #32                                            ;
                                                6B  DD 0001B          PUSHL   (P$)                                           ;
                                                7E  D4 0001D          CLRL    -(SP)                                          ;
                        00000000G               00  FB 0001F          CALLS   #12, SYS$QIOW                                  ;
                                                06  E9 00026          BLBC    STATUS, 1$                                     ; 1700
                                                50  3C 00029          MOVZWL  IOSB, STATUS                                   ;
                                                16  E8 0002C          BLBS    STATUS, 2$                                     ; 1701
                                                50  DD 0002F 1$:      PUSHL   STATUS                                         ; 1703
                     7E         18              AB  20  C1 00031      ADDL3   #32, 24(P$), -(SP)                             ;
                                                    01  DD 00036      PUSHL   #1                                             ;
                                      00000000G     8F  DD 00038      PUSHL   #BACKUP$_WRITEERR+4                            ;
                        00000000G               00  04  FB 0003E      CALLS   #4, LIB$SIGNAL                                 ;
                                                    04 00045 2$:      RET                                                    ; 1704

; Routine Size:  70 bytes,    Routine Base:  CODE + 0295
```

```
;  608          1705  1 %SBTTL 'INIT_BITMAP - initialize storage bitmap'
;  609          1706  1 ROUTINE INIT_BITMAP: L_PS NOVALUE=
;  610          1707  1
;  611          1708  1 !++
;  612          1709  1 !
;  613          1710  1 ! FUNCTIONAL DESCRIPTION:
;  614          1711  1 !       This routine initializes the contents of the volume storage bitmap.
;  615          1712  1 !
;  616          1713  1 ! INPUT PARAMETERS:
;  617          1714  1 !       NONE
;  618          1715  1 !
;  619          1716  1 ! IMPLICIT INPUTS:
;  620          1717  1 !       NONE
;  621          1718  1 !
;  622          1719  1 ! OUTPUT PARAMETERS:
;  623          1720  1 !       NONE
;  624          1721  1 !
;  625          1722  1 ! IMPLICIT OUTPUTS:
;  626          1723  1 !       NONE
;  627          1724  1 !
;  628          1725  1 ! ROUTINE VALUE:
;  629          1726  1 !       NONE
;  630          1727  1 !
;  631          1728  1 ! SIDE EFFECTS:
;  632          1729  1 !       NONE
;  633          1730  1 !
;  634          1731  1 !--
;  635          1732  1
;  636          1733  2 BEGIN
;  637          1734  2 LOCAL
;  638          1735  2         BUFFER:         BBLOCK[512];    ! Block buffer
;  639          1736  2 L_DECL;
;  640          1737  2
;  641          1738  2
;  642          1739  2 ! Build the storage control block and write it out.
;  643          1740  2 !
;  644          1741  2 CH$FILL(0, 512, BUFFER);
;  645          1742  2 IF .STRUCLEV_1
;  646          1743  2 THEN
;  647          1744  3     BEGIN
;  648          1745  3     MAP
;  649          1746  3         BUFFER:         VECTOR;
;  650          1747  3     LOCAL
;  651          1748  3         BLOCK_COUNT;            ! number of blocks in storage map
;  652          1749  3
;  653          1750  3     BLOCK_COUNT = .BITMAP_CNT - 1;
;  654          1751  3     IF .BLOCK_COUNT GTRU 126 THEN BLOCK_COUNT = 0;
;  655          1752  3     (BUFFER+3)<0,8> = .BLOCK_COUNT;
;  656          1753  3     DECR J FROM .BLOCK_COUNT TO 1 DO BUFFER[.J] = 4096;
;  657          1754  3     BUFFER[.BLOCK_COUNT+1] = ROT(.VOLUME_SIZE, 16);
;  658          1755  3     END
;  659          1756  2 ELSE
;  660          1757  3     BEGIN
;  661          1758  3     BUFFER[SCB$W_STRUCLEV] = SCB$C_LEVEL2 + 1;
;  662          1759  3     BUFFER[SCB$W_CLUSTER] = .CLUSTER;
;  663          1760  3     BUFFER[SCB$L_VOLSIZE] = .DEVCHAR[DIB$L_MAXBLOCK];
;  664          1761  4     BUFFER[SCB$L_BLKSIZE] = (.DEVCHAR[DIB$B_SECTORS]
```

```
 4                                  * .DEVCHAR[DIB$B_TRACKS]
 4                                  * .DEVCHAR[DIB$W_CYLINDERS])
 3                                  / .DEVCHAR[DIB$L_MAXBLOCK];
 3         BUFFER[SCB$L_SECTORS] = .DEVCHAR[DIB$B_SECTORS];
 3         BUFFER[SCB$L_TRACKS] = .DEVCHAR[DIB$B_TRACKS];
 3         BUFFER[SCB$L_CYLINDER] = .DEVCHAR[DIB$W_CYLINDERS];
 3         CHECKSUM2(BUFFER, $BYTEOFFSET(SCB$W_CHECKSUM));
 2         END;
 2 WRITE_BLOCK(.BITMAP_LBN, BUFFER);
 2
 2
 2 ! Now initialize the bitmap to binary zeros.  This will also initialize any
 2 ! extra blocks that may be present due to cluster rounding.  The correct
 2 ! contents of the bitmap will be written at the end of the restore operation.
 2 !
 2 CH$FILL(0, 512, BUFFER);
 2 INCRU LBN FROM .BITMAP_LBN + 1 TO .BITMAP_LBN + .BITMAP_CNT - 1 DO
 2     WRITE_BLOCK(.LBN, BUFFER);
 1 END;
```

```
                                           003C 00000 INIT_BITMAP:
                                                            .WORD   Save R2,R3,R4,R5                            ; 1706
                          5E      FE00     CE   9E 00002    MOVAB   -512(SP), SP                                ;
 0200  8F        00       6E               00   2C 00007    MOVC5   #0, (SP), #0, #512, BUFFER                  ; 1741
                                           6E      0000E                                                        ;
                          2C        04     AB   E9 0000F    BLBC    4(P$), 4$                                   ; 1742
                 51    34 AB               01   C3 00013    SUBL3   #1, 52(P$), BLOCK_COUNT                     ; 1750
                0000007E  8F               51   D1 00018    CMPL    BLOCK_COUNT, #126                           ; 1751
                          02               1B   0001F       BLEQU   1$                                          ;
                          51               D4   00021       CLRL    BLOCK_COUNT                                 ;
                       03 AE               51   90 00023 1$: MOVB   BLOCK_COUNT, BUFFER+3                       ; 1752
                          50        01     A1   9E 00027    MOVAB   1(R1), J                                    ; 1753
                          06               11   0002B       BRB     3$                                          ;
                        6E40      1000     8F   3C 0002D 2$: MOVZWL #4096, BUFFER[J]                            ;
                          F7               50   F5 00033 3$: SOBGTR J, 2$                                       ;
            04 AE41    0C AB               10   9C 00036    ROTL    #16, 12(P$), BUFFER+4[BLOCK_COUNT]          ; 1754
                          49               11   0003D       BRB     5$                                          ; 1742
                          6E      0201     8F   B0 0003F 4$: MOVW   #513, BUFFER                                ; 1758
                       02 AE        14     AB   B0 00044    MOVW    20(P$), BUFFER+2                            ; 1759
                          50        1C     AB   D0 00049    MOVL    28(P$), R0                                  ; 1760
                       04 AE        70     A0   D0 0004D    MOVL    112(R0), BUFFER+4                           ;
                          51        08     A0   9A 00052    MOVZBL  8(R0), R1                                   ; 1762
                          52        09     A0   9A 00056    MOVZBL  9(R0), R2                                   ;
                          51               52   C4 0005A    MULL2   R2, R1                                      ;
                          53        0A     A0   3C 0005D    MOVZWL  10(R0), R3                                  ; 1763
                          51               53   C4 00061    MULL2   R3, R1                                      ;
            08    AE      51        70     A0   C7 00064    DIVL3   112(R0), R1, BUFFER+8                       ; 1764
                       0C AE        08     A0   9A 0006A    MOVZBL  8(R0), BUFFER+12                            ; 1765
                       10 AE        09     A0   9A 0006F    MOVZBL  9(R0), BUFFER+16                            ; 1766
                       14 AE        0A     A0   3C 00074    MOVZWL  10(R0), BUFFER+20                           ; 1767
                          7E      01FE     8F   3C 00079    MOVZWL  #510, -(SP)                                 ; 1768
                                    04     AE   9F 0007E    PUSHAB  BUFFER                                      ;
                00000000G 00               02   FB 00081    CALLS   #2, CHECKSUM2                               ;
```

```
                                                  5E  DD 00088 5$:     PUSHL   SP                                      ; 1770
                                              50  AB  DD 0008A         PUSHL   80(P$)                                  ;
                                          FF28    CF  02  FB 0008D         CALLS   #2, WRITE_BLOCK                     ;
    0200    8F              00                6E  00  2C 00092         MOVC5   #0, (SP), #0, #512, BUFFER              ; 1777
                                                  6E      00099                                                         ;
                            50      50        AB  34  AB  C1 0009A         ADDL3   52(P$), 80(P$), R0                  ; 1778
                                              53  FF  A0  9E 000A0         MOVAB   -1(R0), R3                          ;
                                              52  50  AB  D0 000A4         MOVL    80(P$), LBN                         ;
                                                  09  11  000A8         BRB     7$                                  ;
                                            4004  8F  BB  000AA 6$:     PUSHR   #^M<R2,SP>                          ; 1779
                                          FF07    CF  02  FB 000AE         CALLS   #2, WRITE_BLOCK                     ;
                                                  52  D6  000B3 7$:     INCL    LBN                                 ;
                                              53      52  D1  000B5         CMPL    LBN, R3                             ;
                                                  F0  1B  000B8         BLEQU   6$                                  ;
                                                      04  000BA         RET                                         ; 1780

; Routine Size:  187 bytes,    Routine Base:  CODE + 02DB
```

```
;  685          1781  1 %SBTTL 'Initial ODS-1 file header'
;  686          1782  1 BIND
;  687          1783  1         INITIAL_HEADER_1 = UPLIT (
;  688          1784  1                                                  ! HEADER area
;  689          1785  1         BYTE (FH1$C_LENGTH / 2),                 ! ident area offset
;  690          1786  1         BYTE ((FH1$C_LENGTH + FI1$C_LENGTH)/2),  ! map area offset
;  691          1787  1         WORD (FID$C_INDEXF, FID$C_INDEXF),       ! file ID
;  692          1788  1         BYTE (1, 1),                             ! structure version and level
;  693          1789  1         WORD (0),                                ! file owner UIC
;  694          1790  1         WORD (0),                                ! file protection
;  695          1791  1         WORD (0),                                ! file characteristics
;  696          1792  1         BYTE (0),                                ! record type
;  697          1793  1         BYTE (0),                                ! record attributes
;  698          1794  1         WORD (0),                                ! record size
;  699          1795  1         LONG (0, 0),                             ! HIBLK and EFBLK
;  700          1796  1         WORD (0),                                ! EOF byte offset
;  701          1797  1         BYTE (0, 0),                             ! bucket size & VFC length
;  702          1798  1         WORD (0),                                ! maximum record length
;  703          1799  1         WORD (0),                                ! default extend size
;  704          1800  1         WORD (0, 0, 0, 0, 0, 0),                 ! unused record attributes
;  705          1801  1
;  706          1802  1                                                  ! IDENT area
;  707          1803  1         WORD (%RAD50_11'INDEXF   SYS', 1),       ! file name, type and version
;  708          1804  1         WORD (1),                                ! revision number
;  709          1805  1         BYTE (REP 34 OF (0)),                    ! dates
;  710          1806  1
;  711          1807  1                                                  ! MAP area
;  712          1808  1         BYTE (0),                                ! file segment number
;  713          1809  1         BYTE (0),                                ! extension RVN
;  714          1810  1         WORD (0, 0),                             ! extension file ID
;  715          1811  1         BYTE (1, 3),                             ! map pointer count & LBN size
;  716          1812  1         BYTE (0),                                ! map words in use
;  717          1813  1         BYTE ((512-2-FH1$C_LENGTH-FI1$C_LENGTH-FM1$C_LENGTH)/2));
;  718          1814  1                                                  ! map words available
```

```
;  720          1815  1 %SBTTL 'INIT_INDEX1 - initialize ODS-1 index file'
;  721          1816  1 ROUTINE INIT_INDEX1: L_P$ NOVALUE=
;  722          1817  1
;  723          1818  1 !++
;  724          1819  1 !
;  725          1820  1 ! FUNCTIONAL DESCRIPTION:
;  726          1821  1 !       This routine initializes the contents of an ODS-1 index file.
;  727          1822  1 !       It writes the boot block, the home block, and the initial headers.
;  728          1823  1 !
;  729          1824  1 ! INPUT PARAMETERS:
;  730          1825  1 !       NONE
;  731          1826  1 !
;  732          1827  1 ! IMPLICIT INPUTS:
;  733          1828  1 !       NONE
;  734          1829  1 !
;  735          1830  1 ! OUTPUT PARAMETERS:
;  736          1831  1 !       NONE
;  737          1832  1 !
;  738          1833  1 ! IMPLICIT OUTPUTS:
;  739          1834  1 !       NONE
;  740          1835  1 !
;  741          1836  1 ! ROUTINE VALUE:
;  742          1837  1 !       NONE
;  743          1838  1 !
;  744          1839  1 ! SIDE EFFECTS:
;  745          1840  1 !       NONE
;  746          1841  1 !
;  747          1842  1 !--
;  748          1843  1
;  749          1844  2 BEGIN
;  750          1845  2 LOCAL
;  751          1846  2         BUFFER:         BBLOCK[512],    ! Block buffer
;  752          1847  2         ALT_BUFFER:     BBLOCK[512],    ! Block buffer
;  753          1848  2         LBN,                            ! Current LBN
;  754          1849  2         BAD:            REF BBLOCK,     ! Pointer to bad block entry
;  755          1850  2         EXTENSION_FID,                  ! File number for extension headers
;  756          1851  2         IDXFILE_EXT_CNT,                ! Remaining count for the index file
;  757          1852  2         IDXFILE_EXT_LBN,                ! Starting block for the remaining count
;  758          1853  2         MAP_FULL_STATUS,                ! Map area full indicator
;  759          1854  2         UNMAPPED;                       ! Remaining count when header full
;  760          1855  2 LITERAL
;  761          1856  2         IDXFILE_EXT_FID = 6;            ! first index file extension header file number
;  762          1857  2 BIND
;  763          1858  2         IDENT_AREA      = BUFFER + FH1$C_LENGTH : BBLOCK,
;  764          1859  2         MAP_AREA        = BUFFER + FH1$C_LENGTH + FI1$C_LENGTH : BBLOCK,
;  765          1860  2         ALT_MAP_AREA    = ALT_BUFFER + FH1$C_LENGTH + FI1$C_LENGTH : BBLOCK;
;  766          1861  2 L_DECL;
;  767          1862  2
;  768          1863  2
;  769          1864  2 ! First block to write is the boot block.  If a boot block was present on the
;  770          1865  2 ! input volume, write the boot program.  Otherwise, set up the message routine
;  771          1866  2 ! for the -11 and build the message.
;  772          1867  2 !
;  773          1868  2 IF .BBLOCK[OUTPUT_ATTBUF[VSR_BOOTBLOCK], DSC$W_LENGTH] NEQ 0
;  774          1869  2 THEN
;  775          1870  3     BEGIN
;  776          1871  3     CH$COPY(
```

```
;  777          1872  3          .BBLOCK[OUTPUT_ATTBUF[VSR_BOOTBLOCK], DSC$W_LENGTH],
;  778          1873  3          .BBLOCK[OUTPUT_ATTBUF[VSR_BOOTBLOCK], DSC$A_POINTER],
;  779          1874  3          0,
;  780          1875  3          512, BUFFER);
;  781          1876  3      BUFFER[4,0,32,0] = -1;      ! Initialize with invalid LBN
;  782          1877  3      END
;  783          1878  2 ELSE
;  784          1879  3      BEGIN
;  785          1880  3      CH$COPY(
;  786          1881  3          BOOT_PROG_LEN, BOOT_PROGRAM,
;  787          1882  3          BOOT_MESG_LEN, BOOT_MESSAGE,
;  788          1883  3          0,
;  789          1884  3          512, BUFFER);
;  790          1885  3      CH$COPY(
;  791          1886  3          .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH],
;  792          1887  3          .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER],
;  793          1888  3          %C' ',
;  794          1889  3          BTB$S_VOLNAME, BUFFER[BTB$T_VOLNAME]);
;  795          1890  2      END;
;  796          1891  2 WRITE_BLOCK(.BOOTBLOCK_LBN, BUFFER);
;  797          1892  2
;  798          1893  2
;  799          1894  2 ! Now construct and write the home block.
;  800          1895  2 !
;  801          1896  2 CH$FILL(0, 512, BUFFER);
;  802          1897  2 BUFFER[HM1$W_IBMAPSIZE] = (.OUTPUT_ATTBUF[VSR_MAXFILES] + 4095) / 4096;
;  803          1898  2 BUFFER[HM1$L_IBMAPLBN] = ROT(.IDXFILE_LBN, 16);
;  804          1899  2 BUFFER[HM1$W_MAXFILES] = .OUTPUT_ATTBUF[VSR_MAXFILES];
;  805          1900  2 BUFFER[HM1$W_CLUSTER] = 1;
;  806          1901  2 BUFFER[HM1$W_STRUCLEV] = .OUTPUT_ATTBUF[VSR_VOLSTRUCT];
;  807          1902  2 CH$COPY(
;  808          1903  2      .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH],
;  809          1904  2      .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER],
;  810          1905  2      0,
;  811          1906  2      HM1$S_VOLNAME, BUFFER[HM1$T_VOLNAME]);
;  812          1907  2 (BUFFER[HM1$W_VOLOWNER])<0,8> = .(OUTPUT_ATTBUF[VSR_VOLOWNER])<0,8>;
;  813          1908  2 (BUFFER[HM1$W_VOLOWNER])<8,8> = .(OUTPUT_ATTBUF[VSR_VOLOWNER])<16,8>;
;  814          1909  2 BUFFER[HM1$W_PROTECT] = .OUTPUT_ATTBUF[VSR_PROTECT];
;  815          1910  2 BUFFER[HM1$W_FILEPROT] = .OUTPUT_ATTBUF[VSR_FILEPROT];
;  816          1911  2 BUFFER[HM1$B_WINDOW] = .OUTPUT_ATTBUF[VSR_WINDOW];
;  817          1912  2 BUFFER[HM1$B_EXTEND] = .OUTPUT_ATTBUF[VSR_EXTEND];
;  818          1913  2 BUFFER[HM1$B_LRU_LIM] = .OUTPUT_ATTBUF[VSR_LRU_LIM];
;  819          1914  2 TO_ODS1_DATE(OUTPUT_ATTBUF[VSR_VOLDATE], BUFFER[HM1$T_CREDATE]);
;  820          1915  2 BUFFER[HM1$L_SERIALNUM] = .OUTPUT_BAD[BAD_SERIAL];
;  821          1916  2 CH$COPY(
;  822          1917  2      .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH],
;  823          1918  2      .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER],
;  824          1919  2      %C' ',
;  825          1920  2      HM1$S_VOLNAME2, BUFFER[HM1$T_VOLNAME2]);
;  826          1921  2 CH$COPY(
;  827          1922  2      .BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$W_LENGTH],
;  828          1923  2      .BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$A_POINTER],
;  829          1924  2      %C' ',
;  830          1925  2      HM1$S_OWNERNAME, BUFFER[HM1$T_OWNERNAME]);
;  831          1926  2 CH$MOVE(HM1$S_FORMAT, UPLIT BYTE('DECFILE11A  '), BUFFER[HM1$T_FORMAT]);
;  832          1927  2 CHECKSUM2(BUFFER, $BYTEOFFSET(HM1$W_CHECKSUM1));
;  833          1928  2 CHECKSUM2(BUFFER, $BYTEOFFSET(HM1$W_CHECKSUM2));
```

```
;  834          1929  2 WRITE_BLOCK(.HOMEBLOCK1_LBN, BUFFER);
;  835          1930  2
;  836          1931  2
;  837          1932  2 ! Finish initializing the VCB, except for the index file window, which is
;  838          1933  2 ! done after the header is written.
;  839          1934  2 !
;  840          1935  2 VCB[VCB_CLUSTER] = 1;
;  841          1936  2 VCB[VCB_HDR_OFFSET] = 2 + .BUFFER[HM1$W_IBMAPSIZE];
;  842          1937  2 VCB[VCB_MAXFILIDX] = .BUFFER[HM1$W_IBMAPSIZE] * 4096;
;  843          1938  2 VCB[VCB_IMAP_LBN] = .IDXFILE_LBN;
;  844          1939  2
;  845          1940  2
;  846          1941  2 ! Initialize the memory resident index file bitmap.  The first block contains
;  847          1942  2 ! the initially allocated files marked in use; the rest is all zero.
;  848          1943  2 !
;  849          1944  2 VCB[VCB_IMAP] = GET_ZERO_VM(.BUFFER[HM1$W_IBMAPSIZE] * 512);
;  850          1945  2 .VCB[VCB_IMAP] = %B'111';
;  851          1946  2
;  852          1947  2
;  853          1948  2 ! Construct and write the index file header.
;  854          1949  2 !
;  855          1950  2 LBN = .IDXFILE_LBN + .BUFFER[HM1$W_IBMAPSIZE] - 1;
;  856          1951  2 CH$COPY(
;  857          1952  2     FH1$C_LENGTH+FI1$C_LENGTH+FM1$C_LENGTH, INITIAL_HEADER_1,
;  858          1953  2     0,
;  859          1954  2     512, BUFFER);
;  860          1955  2 INCR J FROM BOOTBLOCK_IDX TO IDXFILE_IDX-1 DO
;  861          1956  3     BEGIN
;  862          1957  3     IF .ALLOC_TABLE_CNT[.J] NEQ 0
;  863          1958  3     THEN MAKE_POINTER1(BUFFER, .ALLOC_TABLE_CNT[.J], .ALLOC_TABLE_LBN[.J]);
;  864          1959  2     END;
;  865          1960  2
;  866          1961  2 ! Now that the basic information has been accounted for, account for the
;  867          1962  2 ! space required to keep the index file headers.  This gets interesting
;  868          1963  2 ! if it is necessary to generate extension headers.
;  869          1964  2 !
;  870          1965  2
;  871          1966  2 IDXFILE_EXT_CNT = .ALLOC_TABLE_CNT[IDXFILE_IDX];
;  872          1967  2 IDXFILE_EXT_LBN = .ALLOC_TABLE_LBN[IDXFILE_IDX];
;  873          1968  2
;  874          1969  2 IF NOT MAKE_POINTER1 (BUFFER, .IDXFILE_EXT_CNT, .IDXFILE_EXT_LBN, UNMAPPED)
;  875          1970  2 THEN
;  876          1971  3     BEGIN
;  877          1972  3     EXTENSION_FID = IDXFILE_EXT_FID;
;  878          1973  3     MAP_AREA[FM1$B_EX_RVN] = 0;
;  879          1974  3     MAP_AREA[FM1$W_EX_FILNUM] = .EXTENSION_FID;
;  880          1975  3     MAP_AREA[FM1$W_EX_FILSEQ] = .EXTENSION_FID;
;  881          1976  3     MAP_AREA[FM1$B_EX_SEGNUM] = 0;
;  882          1977  3     CHECKSUM (BUFFER);
;  883          1978  3     WRITE_BLOCK(.LBN + FID$C_INDEXF, BUFFER);
;  884          1979  3
;  885          1980  3     CH$MOVE (512, BUFFER, ALT_BUFFER);
;  886          1981  3     DO
;  887          1982  4         BEGIN
;  888          1983  4         ALT_BUFFER[FH1$W_FID_NUM] = .EXTENSION_FID;
;  889          1984  4         ALT_BUFFER[FH1$W_FID_SEQ] = .EXTENSION_FID;
;  890          1985  4         ALT_MAP_AREA[FM1$B_INUSE] = 0;
```

```
;   891          1986  4          CH$FILL (0, 512-FH1$C_LENGTH-FI1$C_LENGTH-FM1$C_LENGTH,
;   892          1987  4                  ALT_BUFFER+FA1$C_LENGTH+FI1$C_LENGTH+FM1$C_LENGTH);
;   893          1988  4          IDXFILE_EXT_LBN = .IDXFILE_EXT_LBN - .UNMAPPED + .IDXFILE_EXT_CNT;
;   894          1989  4          IDXFILE_EXT_CNT = .UNMAPPED;
;   895          1990  4          MAP_FULL_STATUS = MAKE_POINTER1 (ALT_BUFFER, .IDXFILE_EXT_CNT, .IDXFILE_EXT_LBN, UNMAPPED);
;   896          1991  4          ALT_MAP_AREA[FM1$W_EX_FILNUM] = .EXTENSION_FID + 1;
;   897          1992  4          ALT_MAP_AREA[FM1$W_EX_FILSEQ] = .EXTENSION_FID + 1;
;   898          1993  4          IF .MAP_FULL_STATUS
;   899          1994  4          THEN
;   900          1995  5              BEGIN
;   901          1996  5              ALT_MAP_AREA[FM1$W_EX_FILNUM] = 0;
;   902          1997  5              ALT_MAP_AREA[FM1$W_EX_FILSEQ] = 0;
;   903          1998  4              END;
;   904          1999  4          ALT_MAP_AREA[FM1$B_EX_SEGNUM] = .ALT_MAP_AREA[FM1$B_EX_SEGNUM] + 1;
;   905          2000  4          CHECKSUM (ALT_BUFFER);
;   906          2001  4          WRITE_BLOCK (.LBN + .EXTENSION_FID, ALT_BUFFER);
;   907          2002  4          EXTENSION_FID = .EXTENSION_FID + 1;
;   908          2003  4          END
;   909          2004  3      UNTIL .MAP_FULL_STATUS;
;   910          2005  3
;   911          2006  3  ! Mark any created index file extension headers as in use
;   912          2007  3  !
;   913          2008  3
;   914          2009  3      (.VCB[VCB_IMAP])<IDXFILE_EXT_FID-1,.EXTENSION_FID - IDXFILE_EXT_FID> = -1;
;   915          2010  3      END
;   916          2011  2  ELSE
;   917          2012  3      BEGIN
;   918          2013  3      CHECKSUM (BUFFER);
;   919          2014  3      WRITE_BLOCK (.LBN + FID$C_INDEXF, BUFFER);
;   920          2015  2      END;
;   921          2016  2
;   922          2017  2  CREATE_WINDOW(BUFFER, 1, VCB[VCB_INDEXF], 1, 0);
;   923          2018  2
;   924          2019  2
;   925          2020  2  ! Construct and write the bad block file header.
;   926          2021  2  !
;   927          2022  2  CH$FILL(0, 512-FH1$C_LENGTH-FI1$C_LENGTH-FM1$C_LENGTH, BUFFER+FH1$C_LENGTH+FI1$C_LENGTH+FM1$C_LENGTH);
;   928          2023  2  BUFFER[FH1$W_FID_NUM] = FID$C_BADBLK;
;   929          2024  2  BUFFER[FH1$W_FID_SEQ] = FID$C_BADBLK;
;   930          2025  2  MAP_AREA[FM1$W_EX_FILNUM] = 0;
;   931          2026  2  MAP_AREA[FM1$W_EX_FILSEQ] = 0;
;   932          2027  2  MAP_AREA[FM1$B_EX_SEGNUM] = 0;
;   933          2028  2  MAP_AREA[FM1$B_INUSE] = 0;
;   934          2029  2  (IDENT_AREA[FIT$W_FILENAME])<0,32> = %RAD50_11'BADBLK';
;   935          2030  2  BAD = OUTPUT_BAD[BAD_DESC];
;   936          2031  2  INCR J FROM 0 TO .OUTPUT_BAD[BAD_NUMDESC]-1 DO
;   937          2032  3      BEGIN
;   938          2033  3      IF NOT MAKE_POINTER1(BUFFER, .BAD[BAD_COUNT], .BAD[BAD_LBN])
;   939          2034  3      THEN SIGNAL(BACKUP$_MAXBAD, 1, VCB[VCB_DEVICE]);
;   940          2035  3      BAD = .BAD + BAD_S_DESC;
;   941          2036  2      END;
;   942          2037  2  CHECKSUM(BUFFER);
;   943          2038  2  WRITE_BLOCK(.LBN + FID$C_BADBLK, BUFFER);
;   944          2039  2
;   945          2040  2
;   946          2041  2  ! Construct and write the storage map file header.
;   947          2042  2  !
```

```
;  948          2043  2 CH$FILL (0, 512-FH1$C_LENGTH-FI1$C_LENGTH-FM1$C_LENGTH, BUFFER+FH1$C_LENGTH+FI1$C_LENGTH+FM1$C_LENGTH);
;  949          2044  2 BUFFER[FH1$W_FID_NUM] = FID$C_BITMAP;
;  950          2045  2 BUFFER[FH1$W_FID_SEQ] = FID$C_BITMAP;
;  951          2046  2 MAP_AREA[FM1$B_INUSE] = 0;
;  952          2047  2 (IDENT_AREA[FIT$W_FILENAME])<0,32> = %RAD50_11'BITMAP';
;  953          2048  2 MAKE_POINTER1(BUFFER, 1, .BITMAP_LBN);
;  954          2049  2 MAKE_POINTER1(BUFFER, .BITMAP_CNT-1, .BITMAP_LBN+1);
;  955          2050  2 CHECKSUM(BUFFER);
;  956          2051  2 WRITE_BLOCK(.LBN + FID$C_BITMAP, BUFFER);
;  957          2052  1 END;


                                                00396          .BLKB   2
                                             17 00398 P.AAC:   .BYTE   23                                                    ;
                                             2E 00399          .BYTE   46                                                    ;
                                      0001   0001 0039A        .WORD   1, 1                                                  ;
                                         01   01 0039E         .BYTE   1, 1                                                  ;
                                           0000 003A0          .WORD   0                                                     ;
                                           0000 003A2          .WORD   0                                                     ;
                                           0000 003A4          .WORD   0                                                     ;
                                             00 003A6          .BYTE   0                                                     ;
                                             00 003A7          .BYTE   0                                                     ;
                                           0000 003A8          .WORD   0                                                     ;
                                 00000000  00000000 003AA      .LONG   0, 0                                                  ;
                                           0000 003B2          .WORD   0                                                     ;
                                        00   00 003B4          .BYTE   0, 0                                                  ;
                                           0000 003B6          .WORD   0                                                     ;
                                           0000 003B8          .WORD   0                                                     ;
                  0000  0000   0000  0000  0000  0000 003BA    .WORD   0, 0, 0, 0, 0, 0                                      ;
                      7A  BB   00  00  23  06  3A  74 003C6     .RAD50  \INDEXF   SYS\                                        ;
                                           0001 003CE          .WORD   1                                                     ;
                                           0001 003D0          .WORD   1                                                     ;
                                             00# 003D2         .BYTE   0[34]                                                 ;
                                             00 003F4          .BYTE   0                                                     ;
                                             00 003F5          .BYTE   0                                                     ;
                                      0000   0000 003F6        .WORD   0, 0                                                  ;
                                         03   01 003FA         .BYTE   1, 3                                                  ;
                                             00 003FC          .BYTE   0                                                     ;
                                             CC 003FD          .BYTE   -52                                                   ;
          20  20  41  31  31  45  4C  49  46  43  45  44 003FE P.AAD:   .ASCII  \DECFILE11A  \                                ;

                                                            INITIAL_HEADER_1=   P.AAC


                                                07FC 00000 INIT_INDEX1:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                          ; 1816
                                    5E     FBFC   CE  9E 00002     MOVAB   -1028(SP), SP                                    ;
                                    50 00000000'  EF  3C 00007     MOVZWL  OUTPUT_ATTBUF+88, R0                             ; 1868
                                                  15  13 0000E     BEQL    1$                                               ;
  0200  8F          00 00000000'  FF              50  2C 00010     MOVC5   R0, @OUTPUT_ATTBUF+92, #0, #512, BUFFER          ; 1871
                                          FE00    CD     0001B                                                             ;
                            FE04  CD              01  CE 0001E     MNEGL   #1, BUFFER+4                                     ; 1876
                                                  22  11 00023     BRB     2$                                               ; 1868
                 FE00   CD     FBCC   CF          26  28 00025 1$:  MOVC3   #38, BOOT_PROGRAM, BUFFER                       ; 1880
  01DA  8F          00     FBEA   CF              28  2C 0002D     MOVC5   #40, BOOT_MESSAGE, #0, #474, (R3)                ;
                                                  63     00036                                                             ;
```

```
                0C      20 00000000'    FF 00000000'    EF  2C 00037            MOVC5   OUT UT_ATTBUF, @OUTPUT_ATTBUF+4, #32, #12, -;  1889
                                                FE26    CD      00044                   BUFFER+38
                                                FE00    CD  9F  00047 2$:       PUSHAB  BUFFER                                          ;  1891
                                                  3C    AB  DD  0004B           PUSHL   60(P$)                                          ;
                      FE38      CF                    02  FB  0004E           CALLS   #2, WRITE_BLOCK                                 ;
          0200  8F          00      6E                00  2C  00053           MOVC5   #0, (SP), #0, #512, BUFFER                      ;  1896
                                            FE00    CD      0005A                                                                   ;
                    50 00000000'    EF  00000FFF    8F  C1  0005D           ADDL3   #4095, OUTPUT_ATTBUF+52, R0                     ;  1897
                    51              50  00001000    8F  C7  00069           DIVL3   #4096, R0, R1                                   ;
                        FE00    CD              51  B0  00071           MOVW    R1, BUFFER                                      ;
                  FE02  CD        4C    AB        10  9C  00076           ROTL    #16, 76(P$), BUFFER+2                           ;  1898
                        FE06    CD  00000000'   EF  B0  0007D           MOVW    OUTPUT_ATTBUF+52, BUFFER+6                      ;  1899
                        FE08    CD              01  B0  00086           MOVW    #1, BUFFER+8                                    ;  1900
                        FE0C    CD  00000000'   EF  B0  0008B           MOVW    OUTPUT_ATTBUF+64, BUFFER+12                     ;  1901
                0C          00 00000000'    FF 00000000'    EF  2C 00094        MOVC5   OUTPUT_ATTBUF, @OUTPUT_ATTBUF+4, #0, #12, -     ;  1906
                                                FE0E    CD      000A1                   BUFFER+14                                       ;
                        FE1E    CD 00000000'    EF  90  000A4           MOVB    OUTPUT_ATTBUF+40, BUFFER+30                     ;  1907
                        FE1F    CD 00000000'    EF  90  000AD           MOVB    OUTPUT_ATTBUF+42, BUFFER+31                     ;  1908
                        FE20    CD 00000000'    EF  B0  000B6           MOVW    OUTPUT_ATTBUF+68, BUFFER+32                     ;  1909
                        FE24    CD 00000000'    EF  B0  000BF           MOVW    OUTPUT_ATTBUF+70, BUFFER+36                     ;  1910
                        FE2C    CD 00000000'    EF  90  000C8           MOVB    OUTPUT_ATTBUF+82, BUFFER+44                     ;  1911
                        FE2D    CD 00000000'    EF  90  000D1           MOVB    OUTPUT_ATTBUF+76, BUFFER+45                     ;  1912
                        FE2E    CD 00000000'    EF  90  000DA           MOVB    OUTPUT_ATTBUF+83, BUFFER+46                     ;  1913
                                            FE3C    CD  9F  000E3           PUSHAB  BUFFER+60                                       ;  1914
                                            00000000'   EF  9F  000E7           PUSHAB  OUTPUT_ATTBUF+24                                ;
                    00000000G   0C                  02  FB  000ED           CALLS   #2, TO_ODS1_DATE                                ;
                                            50  00000000'   EF  D0  000F4           MOVL    OUTPUT_BAD, R0                          ;  1915
                      C8    AD        04    A0  D0  000FB           MOVL    4(R0), BUFFER+456                               ;
                0C          20 00000000'    FF 00000000'    EF  2C 00100        MOVC5   OUTPUT_ATTBUF, @OUTPUT_ATTBUF+4, #32, #12, -    ;  1920
                                                  D8    AD      0010D                   BUFFER+4+2                                      ;
                0C          20 00000000'    FF 00000000'    EF  2C 0010F        MOVC5   OUTPUT_ATTBUF+8, @OUTPUT_ATTBUF+12, #32, -      ;  1925
                                                  E4    AD      0011C                   #12, BUFFER+484                                 ;
              F0  AD      FED1    CF                    0C  28  0011E           MOVC3   #12, P.AAD, BUFFER+496                          ;  1926
                                                          3A  DD  00125           PUSHL   #58                                             ;  1927
                                            FE00    CD      9F  00127           PUSHAB  BUFFER                                          ;
                    00000000G   00                  02  FB  0012B           CALLS   #2, CHECKSUM2                                   ;
                                  7E    01FE    8F      3C  00132           MOVZWL  #510, -(SP)                                     ;  1928
                                            FE00    CD      9F  00137           PUSHAB  BUFFER                                          ;
                    00000000G   00                  02  FB  0013B           CALLS   #2, CHECKSUM2                                   ;
                                            FE00    CD      9F  00142           PUSHAB  BUFFER                                          ;  1929
                                              40    AB      DD  00146           PUSHL   64(P$)                                          ;
                      FD3D      CF                    02  FB  00149           CALLS   #2, WRITE_BLOCK                                 ;
                                  52        18    AB      D0  0014E           MOVL    24(P$), R2                                      ;  1935
                                    04      A2      01  B0  00152           MOVW    #1, 4(R2)                                       ;
                                  53    FE00    CD      3C  00156           MOVZWL  BUFFER, R3                                      ;  1936
                  1A  A2                  53              02  A1  0015B           ADDW3   #2, R3, 26(R2)                                  ;
                  1C  A2                  53              0C  78  00160           ASHL    #12, R3, 28(R2)                                 ;  1937
                                    14    A2        4C    AB  D0  00165           MOVL    76(P$), 20(R2)                                  ;  1938
                        7E                  53              09  78  0016A           ASHL    #9, R3, -(SP)                                   ;  1944
                    00000000G   00                  01  FB  0016E           CALLS   #1, GET_ZERO_VM                                 ;
                                    10      A2      50  D0  00175           MOVL    R0, @16(R2)                                     ;
                                    10      B2      07  D0  00179           MOVL    #7, @16(R2)                                     ;  1945
                        57                  53        4C    AB  C1  0017D           ADDL3   76(P$), R3, R7                                  ;  1950
                                                          57  D7  00182           DECL    LBN                                             ;
          0200  8F          00      FE03    CF      0066    8F  2C  00184       MOVC5   #102, INITIAL_HEADER_1, #0, #512, BUFFER        ;  1951
                                            FE00    CD      0018F                                                                   ;
                                  53        20    AB      9E  00192           MOVAB   32(P$), R3                                      ;  1957
```

```
                                                52  D4 00196        CLRL    J                                        :
                                              6342  D5 00198 3$:    TSTL    (R3)[J]                                  :
                                                12  13 0019B        BEQL    4$                                       :
                                          3C  AB42  DD 0019D        PUSHL   60(P$)[J]                                : 1958
                                              6342  DD 001A1        PUSHL   (R3)[J]                                  :
                                        FE00    CD  9F 001A4        PUSHAB  BUFFER                                   :
                        00000000G   00          03  FB 001A8        CALLS   #3, MAKE_POINTER1                        :
                    E5              52          03  F3 001AF 4$:    AOBLEQ  #3, J, 3$                                : 1955
                                    59      10  A3  D0 001B3        MOVL    16(R3), IDXFILE_EXT_CNT                  : 1966
                                    58      4C  AB  D0 001B7        MOVL    76(P$), IDXFILE_EXT_LBN                  : 1967
                                    52      01  A7  9E 001BB        MOVAB   1(R7), R2                                : 1978
                                          4100  8F  BB 001BF        PUSHR   #^M<R8,SP>                               : 1969
                                                59  DD 001C3        PUSHL   IDXFILE_EXT_CNT                          :
                                        FE00    CD  9F 001C5        PUSHAB  BUFFER                                   :
                        00000000G   00          04  FB 001C9        CALLS   #4, MAKE_POINTER1                        :
                                    03          50  E9 001D0        BLBC    R0, 5$                                   :
                                              00A4  31 001D3        BRW     8$                                       :
                                    56          06  D0 001D6 5$:    MOVL    #6, EXTENSION_FID                        : 1972
                                        FE5C    CD  B4 001D9        CLRW    MAP_AREA                                 : 1976
                        FE5E        CD          56  B0 001DD        MOVW    EXTENSION_FID, MAP_AREA+2                : 1974
                        FE60        CD          56  B0 001E2        MOVW    EXTENSION_FID, MAP_AREA+4                : 1975
                                        FE00    CD  9F 001E7        PUSHAB  BUFFER                                   : 1977
                        00000000G   00          01  FB 001EB        CALLS   #1, CHECKSUM                             :
                                        FE00    CD  9F 001F2        PUSHAB  BUFFER                                   : 1978
                                                52  DD 001F6        PUSHL   R2                                       :
                        FC8E        CF          02  FB 001F8        CALLS   #2, WRITE_BLOCK                          :
               04   AE  FE00        CD    0200  8F  28 001FD        MOVC3   #512, BUFFER, ALT_BUFFER                 : 1980
                          06        AE          56  B0 00206 6$:    MOVW    EXTENSION_FID, ALT_BUFFER+2              : 1983
                          08        AE          56  B0 0020A        MOVW    EXTENSION_FID, ALT_BUFFER+4              : 1984
                                            68  AE  94 0020E        CLRB    ALT_MAP_AREA+8                           : 1985
 019A  8F           00              6E          00  2C 00211        MOVC5   #0, (SP), #0, #410, ALT_BUFFER+10?       : 1987
                                            6A  AE     00218                                                         :
                    50              58          6E  C3 0021A        SUBL3   UNMAPPED, IDXFILE_EXT_LBN, R0            : 1988
                    58              50          59  C1 0021E        ADDL3   IDXFILE_EXT_CNT, R0, IDXFILE_EXT_LBN     :
                                    59          6E  D0 00222        MOVL    UNMAPPED, IDXFILE_EXT_CNT                : 1989
                                          4100  8F  BB 00225        PUSHR   #^M<R8,SP>                               : 1990
                                                59  DD 00229        PUSHL   IDXFILE_EXT_CNT                          :
                                            10  AE  9F 0022B        PUSHAB  ALT_BUFFER                               :
                        00000000G   00          04  FB 0022E        CALLS   #4, MAKE_POINTER1                        :
                                    5A          50  D0 00235        MOVL    R0, MAP_FULL_STATUS                      :
                                    50      01  A6  9E 00238        MOVAB   1(R6), R0                                : 1991
                          62        AE          50  B0 0023C        MOVW    R0, ALT_MAP_AREA+2                       :
                          64        AE          50  B0 00240        MOVW    R0, ALT_MAP_AREA+4                       : 1992
                                    03          5A  E9 00244        BLBC    MAP_FULL_STATUS, 7$                      : 1993
                                            62  AE  D4 00247        CLRL    ALT_MAP_AREA+2                           : 1996
                                            60  AE  96 0024A 7$:    INCB    ALT_MAP_AREA                             : 1999
                                            04  AE  9F 0024D        PUSHAB  ALT_BUFFER                               : 2000
                        00000000G   00          01  FB 00250        CALLS   #1, CHECKSUM                             :
                                            04  AE  9F 00257        PUSHAB  ALT_BUFFER                               : 2001
                                              6647  9F 0025A        PUSHAB  (EXTENSION_FID)[LBN]                     :
                        FC29        CF          02  FB 0025D        CALLS   #2, WRITE_BLOCK                          :
                                                56  D6 00262        INCL    EXTENSION_FID                            : 2002
                                    9F          5A  E9 00264        BLBC    MAP_FULL_STATUS, 6$                      : 2004
                                    50      18  AB  D0 00267        MOVL    24(P$), R0                               : 2009
                                    56          06  C2 0026B        SUBL2   #6, R6                                   :
   10  B0           56              05 FFFFFFFF 8F  F0 0026E        INSV    #-1, #5, R6, @16(R0)                     :
                                                16  11 00278        BRB     9$                                       : 1969
```

```
                                        FE00    CD  9F 0027A 8$:     PUSHAB  BUFFER                                  ; 2013
                          00000000G     00      01  FB 0027E       CALLS   #1, CHECKSUM
                                        FE00    CD  9F 00285       PUSHAB  BUFFER                                  ; 2014
                                                52  DD 00289       PUSHL   R2
                              FBFB      CF              02  FB 0028B       CALLS   #2, WRITE_BLOCK
                                        7E              01  7D 00290 9$:     MOVQ    #1, -(SP)                       ; 2017
                                              18        AB  DD 00293       PUSHL   24(P$)
                                                        01  DD 00296       PUSHL   #1
                                        FE00    CD  9F 00298       PUSHAB  BUFFER
                          00000000G     00      05  FB 0029C       CALLS   #5, CREATE_WINDOW
019A  8F                        00      6E              00  2C 002A3       MOVC5   #0, (SP), #0, #410, BUFFER+102  ; 2022
                                      FE66      CD      002AA
                              FE02      CD 00030003     8F  D0 002AD       MOVL    #196611, BUFFER+2               ; 2023
                                        FE5E    CD  D4 002B6       CLRL    MAP_AREA+2                              ; 2025
                                        FE5C    CD  94 002BA       CLRB    MAP_AREA                                ; 2027
                                        FE64    CD  94 002BE       CLRB    MAP_AREA+8                              ; 2028
                              FE2E      CD 0E6B0CAC     8F  D0 002C2       MOVL    #247896620, IDENT_AREA          ; 2029
                          52 00000000'  EF              08  C1 002CB       ADDL3   #8, OUTPUT_BAD, BAD             ; 2030
                                        54 00000000'    FF  D0 002D3       MOVL    @OUTPUT_BAD, R4                 ; 2031
                                        53              01  CE 002DA       MNEGL   #1, J
                                                        2A  11 002DD       BRB     12$
                                                        62  DD 002DF 10$:     PUSHL   (BAD)                           ; 2033
                                              04        A2  DD 002E1       PUSHL   4(BAD)
                                        FE00    CD  9F 002E4       PUSHAB  BUFFER
                          00000000G     00      03  FB 002E8       CALLS   #3, MAKE_POINTER1
                                        14              50  E8 002EF       BLBS    R0, 11$
                          7E        18  AB              20  C1 002F2       ADDL3   #32, 24(P$), -(SP)              ; 2034
                                                        01  DD 002F7       PUSHL   #1
                                  00000000G     8F      DD 002F9       PUSHL   #BACKUP$_MAXBAD
                          00000000G     00      03  FB 002FF       CALLS   #3, LIB$SIGNAL
                                        52              08  C0 00306 11$:     ADDL2   #8, BAD                         ; 2035
                          D2            53              54  F2 00309 12$:     AOBLSS  R4, J, 10$                      ; 2031
                                        FE00    CD  9F 0030D       PUSHAB  BUFFER                                  ; 2037
                          00000000G     00      01  FB 00311       CALLS   #1, CHECKSUM
                                        FE00    CD  9F 00318       PUSHAB  BUFFER                                  ; 2038
                                          03    A7  9F 0031C       PUSHAB  3(LBN)
                              FB67      CF              02  FB 0031F       CALLS   #2, WRITE_BLOCK
019A  8F                        00      6E              00  2C 00324       MOVC5   #0, (SP), #0, #410, BUFFER+102  ; 2043
                                      FE66      CD      0032B
                              FE02      CD 00020002     8F  D0 0032E       MOVL    #131074, BUFFER+2               ; 2044
                                        FE64    CD  94 00337       CLRB    MAP_AREA+8                              ; 2046
                              FE2E      CD 517[illegible]DFC    8F  D0 0033B       MOVL    #1366822396, IDENT_AREA ; 2047
                                          50    AB  DD 00344       PUSHL   80(P$)                                  ; 2048
                                                        01  DD 00347       PUSHL   #1
                                        FE00    CD  9F 00349       PUSHAB  BUFFER
                          00000000G     00      03  FB 0034D       CALLS   #3, MAKE_POINTER1
                          7E            50      AB  01  C1 00354       ADDL3   #1, 80(P$), -(SP)                       ; 2049
                          7E            34      AB  01  C3 00359       SUBL3   #1, 52(P$), -(SP)
                                        FE00    CD  9F 0035E       PUSHAB  BUFFER
                          00000000G     00      03  FB 00362       CALLS   #3, MAKE_POINTER1
                                        FE00    CD  9F 00369       PUSHAB  BUFFER                                  ; 2050
                          00000000G     00      01  FB 0036D       CALLS   #1, CHECKSUM
                                        FE00    CD  9F 00374       PUSHAB  BUFFER                                  ; 2051
                                          02    A7  9F 00378       PUSHAB  2(LBN)
                              FB0B      CF              02  FB 0037B       CALLS   #2, WRITE_BLOCK
                                                        04  00380       RET                                     ; 2052
```

; Routine Size: 897 bytes,   Routine Base: CODE + 040A

```
;  959          2053  1 %SBTTL 'Initial ODS-2 file header'
;  960          2054  1 ! Initial file header.  The pending bad block log file is used
;  961          2055  1 ! since it is the first one written. Note that this PLIT must
;  962          2056  1 ! be updated whenever fields are added to the file header.
;  963          2057  1 !
;  964          2058  1 $ASSUME (FH2$C_LENGTH, EQL, 80)
;  965          2059  1 $ASSUME (FI2$C_LENGTH, EQL, 120)
;  966          2060  1
;  967          2061  1 BIND
;  968          2062  1         INITIAL_HEADER  = UPLIT (
;  969          2063  1                                                         ! HEADER area
;  970          2064  1         BYTE (FH2$C_LENGTH / 2),                        ! ident area offset
;  971          2065  1         BYTE ((FH2$C_LENGTH + FI2$C_LENGTH)/2),         ! map area offset
;  972          2066  1         BYTE ($BYTEOFFSET (FH2$W_CHECKSUM)/2),          ! access control list offset
;  973          2067  1         BYTE ($BYTEOFFSET (FH2$W_CHECKSUM)/2),          ! reserved area offset
;  974          2068  1         WORD (0),                                       ! file segment number
;  975          2069  1         BYTE (1, 2),                                    ! structure version and level
;  976          2070  1         WORD (FID$C_BADLOG, FID$C_BADLOG, 0),           ! file ID
;  977          2071  1         WORD (0, 0, 0),                                 ! extension file ID
;  978          2072  1         BYTE (FAT$C_FIXED),                             ! fixed length record type
;  979          2073  1         BYTE (0),                                       ! no record attributes
;  980          2074  1         WORD (16),                                      ! record size
;  981          2075  1         LONG (0, 1^16),                                 ! HIBLK and EFBLK
;  982          2076  1         WORD (0),                                       ! EOF byte offset
;  983          2077  1         BYTE (0, 0),                                    ! bucket size & VFC length
;  984          2078  1         WORD (16),                                      ! maximum record length
;  985          2079  1         WORD (0),                                       ! default extend size
;  986          2080  1         WORD (0, 0, 0, 0, 0, 0),                        ! unused record attributes
;  987          2081  1         LONG (0),                                       ! file characteristics
;  988          2082  1         WORD (0),                                       ! record protection
;  989          2083  1         BYTE (0, 0),                                    ! mapwords in use & access mode
;  990          2084  1         LONG (0),                                       ! file owner UIC
;  991          2085  1         WORD (0),                                       ! file protection
;  992          2086  1         WORD (FID$C_MFD, FID$C_MFD, 0),                 ! directory back link
;  993          2087  1         WORD (0),                                       ! journal control flags
;  994          2088  1         WORD (0),                                       ! spare
;  995          2089  1         LONG (1),                                       ! high water mark
;  996          2090  1
;  997          2091  1                                                         ! IDENT area
;  998          2092  1         BYTE ('BADLOG.SYS;1         '),                 ! file name, type and version
;  999          2093  1         WORD (1),                                       ! revision number
; 1000          2094  1         LONG (0, 0, 0, 0, 0, 0, 0, 0),                  ! dates
; 1001          2095  1         REP FI2$S_FILENAMEXT OF BYTE (' ')              ! file name extension
; 1002          2096  1         );
```

```
;1004            2097  1 %SBTTL 'INIT_INDEX - initialize ODS-2 index file'
;1005            2098  1 ROUTINE INIT_INDEX: L_PS NOVALUE=
;1006            2099  1
;1007            2100  1 !++
;1008            2101  1 !
;1009            2102  1 ! FUNCTIONAL DESCRIPTION:
;1010            2103  1 !       This routine initializes the contents of an ODS-2 index file.
;1011            2104  1 !       It writes the boot block, the home blocks, and the initial headers.
;1012            2105  1 !
;1013            2106  1 ! INPUT PARAMETERS:
;1014            2107  1 !       NONE
;1015            2108  1 !
;1016            2109  1 ! IMPLICIT INPUTS:
;1017            2110  1 !       NONE
;1018            2111  1 !
;1019            2112  1 ! OUTPUT PARAMETERS:
;1020            2113  1 !       NONE
;1021            2114  1 !
;1022            2115  1 ! IMPLICIT OUTPUTS:
;1023            2116  1 !       NONE
;1024            2117  1 !
;1025            2118  1 ! ROUTINE VALUE:
;1026            2119  1 !       NONE
;1027            2120  1 !
;1028            2121  1 ! SIDE EFFECTS:
;1029            2122  1 !       NONE
;1030            2123  1 !
;1031            2124  1 !--
;1032            2125  1
;1033            2126  2 BEGIN
;1034            2127  2 LOCAL
;1035            2128  2         BUFFER:         BBLOCK[512],    ! Block buffer
;1036            2129  2         LBN,                            ! Current LBN
;1037            2130  2         MAP_COUNT,                      ! Count field of map pointer
;1038            2131  2         MAP_LBN,                        ! Start LBN of current map pointer
;1039            2132  2         BAD:            REF BBLOCK;     ! Pointer to bad block entry
;1040            2133  2 BIND
;1041            2134  2         IDENT_AREA      = BUFFER + FH2$C_LENGTH : BBLOCK;
;1042            2135  2 L_DECL;
;1043            2136  2
;1044            2137  2
;1045            2138  2 ! First block to write is the boot block.  If a boot block was present on the
;1046            2139  2 ! input volume, write the boot program.  Otherwise, set up the message routine
;1047            2140  2 ! for the -11 and build the message.
;1048            2141  2 !
;1049            2142  2 IF .BBLOCK[OUTPUT_ATTBUF[VSR_BOOTBLOCK], DSC$W_LENGTH] NEQ 0
;1050            2143  2 THEN
;1051            2144  3     BEGIN
;1052            2145  3     CH$COPY(
;1053            2146  3         .BBLOCK[OUTPUT_ATTBUF[VSR_BOOTBLOCK], DSC$W_LENGTH],
;1054            2147  3         .BBLOCK[OUTPUT_ATTBUF[VSR_BOOTBLOCK], DSC$A_POINTER],
;1055            2148  3         0,
;1056            2149  3         512, BUFFER);
;1057            2150  3     BUFFER[4,0,32,0] = -1;      ! Initialize with invalid LBN
;1058            2151  3     END
;1059            2152  2 ELSE
;1060            2153  3     BEGIN
```

```
;  1061          2154  3     CH$COPY(
;  1062          2155  3         BOOT_PROG_LEN, BOOT_PROGRAM,
;  1063          2156  3         BOOT_MESG_LEN, BOOT_MESSAGE,
;  1064          2157  3         0,
;  1065          2158  3         512, BUFFER);
;  1066          2159  3     CH$COPY(
;  1067          2160  3         .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH],
;  1068          2161  3         .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER],
;  1069          2162  3         %C' ',
;  1070          2163  3         BTB$S_VOLNAME, BUFFER[BTB$T_VOLNAME]);
;  1071          2164  2     END;
;  1072          2165  2 WRITE_BLOCK(.BOOTBLOCK_LBN, BUFFER);
;  1073          2166  2
;  1074          2167  2
;  1075          2168  2 ! Now construct the home block and write it to the remainder of the boot
;  1076          2169  2 ! block cluster and to the two home block clusters.
;  1077          2170  2 !
;  1078          2171  2 CH$FILL(0, 512, BUFFER);
;  1079          2172  2 BUFFER[HM2$L_HOMELBN] = .BOOTBLOCK_LBN + 1;
;  1080          2173  2 BUFFER[HM2$L_ALHOMELBN] = .REAL_HOMEBLOCK;
;  1081          2174  2 BUFFER[HM2$L_ALTIDXLBN] = .IDXHDR2_LBN;
;  1082          2175  2 BUFFER[HM2$W_STRUCLEV] = .OUTPUT_ATTBUF[VSR_VOLSTRUCT];
;  1083          2176  2 BUFFER[HM2$W_CLUSTER] = .CLUSTER;
;  1084          2177  2 BUFFER[HM2$W_HOMEVBN] = 2;
;  1085          2178  2 BUFFER[HM2$W_ALHOMEVBN] = .REAL_HOMEBLOCK - .HOMEBLOCK2_LBN + .CLUSTER * 2 + 1;
;  1086          2179  2 BUFFER[HM2$W_ALTIDXVBN] = .CLUSTER * 3 + 1;
;  1087          2180  2 BUFFER[HM2$W_IBMAPVBN] = .CLUSTER * 4 + 1;
;  1088          2181  2 BUFFER[HM2$L_IBMAPLBN] = .IDXFILE_LBN;
;  1089          2182  2 BUFFER[HM2$L_MAXFILES] = .OUTPUT_ATTBUF[VSR_MAXFILES];
;  1090          2183  2 BUFFER[HM2$W_IBMAPSIZE] = (.OUTPUT_ATTBUF[VSR_MAXFILES] + 4095) / 4096;
;  1091          2184  2 BUFFER[HM2$W_RESFILES] = 9;
;  1092          2185  2 BUFFER[HM2$W_RVN] = .OUTPUT_ATTBUF[VSR_RVN];
;  1093          2186  2 IF .OUTPUT_ATTBUF[VSR_RVN] EQL 1
;  1094          2187  2 THEN
;  1095          2188  2     BUFFER[HM2$W_SETCOUNT] = .COM_O_SETCOUNT;
;  1096          2189  2 BUFFER[HM2$W_VOLCHAR] = .OUTPUT_ATTBUF[VSR_VOLCHAR];
;  1097          2190  2 BUFFER[HM2$L_VOLOWNER] = .OUTPUT_ATTBUF[VSR_VOLOWNER];
;  1098          2191  2 BUFFER[HM2$W_PROTECT] = .OUTPUT_ATTBUF[VSR_PROTECT];
;  1099          2192  2 BUFFER[HM2$W_FILEPROT] = .OUTPUT_ATTBUF[VSR_FILEPROT];
;  1100          2193  2 BUFFER[HM2$W_RECPROT] = .OUTPUT_ATTBUF[VSR_RECPROT];
;  1101          2194  2 (BUFFER[HM2$Q_CREDATE]) = .(OUTPUT_ATTBUF[VSR_VOLDATE]);
;  1102          2195  2 (BUFFER[HM2$Q_CREDATE]+4) = .(OUTPUT_ATTBUF[VSR_VOLDATE]+4);
;  1103          2196  2 BUFFER[HM2$B_WINDOW] = .OUTPUT_ATTBUF[VSR_WINDOW];
;  1104          2197  2 BUFFER[HM2$B_LRU_LIM] = .OUTPUT_ATTBUF[VSR_LRU_LIM];
;  1105          2198  2 BUFFER[HM2$W_EXTEND] = .OUTPUT_ATTBUF[VSR_EXTEND];
;  1106          2199  2 (BUFFER[HM2$Q_RETAINMIN]) = .(OUTPUT_ATTBUF[VSR_RETAINMIN]);
;  1107          2200  2 (BUFFER[HM2$Q_RETAINMIN]+4) = .(OUTPUT_ATTBUF[VSR_RETAINMIN]+4);
;  1108          2201  2 (BUFFER[HM2$Q_RETAINMAX]) = .(OUTPUT_ATTBUF[VSR_RETAINMAX]);
;  1109          2202  2 (BUFFER[HM2$Q_RETAINMAX]+4) = .(OUTPUT_ATTBUF[VSR_RETAINMAX]+4);
;  1110          2203  2 BUFFER[HM2$L_SERIALNUM] = .OUTPUT_BAD[BAD_SERIAL];
;  1111          2204  2 CH$FILL(%C' ', HM2$S_STRUCNAME, BUFFER[HM2$T_STRUCNAME]);
;  1112          2205  2 IF .OUTPUT_ATTBUF[VSR_RVN] NEQ 0
;  1113          2206  2 THEN
;  1114          2207  2     CH$MOVE(HM2$S_STRUCNAME, COM_O_STRUCNAME, BUFFER[HM2$T_STRUCNAME]);
;  1115          2208  2 CH$COPY(
;  1116          2209  2     .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH],
;  1117          2210  2     .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER],
```

```
;  1118          2211  2      %C' ',
;  1119          2212  2      HM2$S_VOLNAME, BUFFER[HM2$T_VOLNAME]);
;  1120          2213  2  CH$COPY(
;  1121          2214  2      .BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$W_LENGTH],
;  1122          2215  2      .BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$A_POINTER],
;  1123          2216  2      %C' ',
;  1124          2217  2      HM2$S_OWNERNAME, BUFFER[HM2$T_OWNERNAME]);
;  1125          2218  2  CH$MOVE(HM2$S_FORMAT, UPLIT BYTE('DECFILE11B  '), BUFFER[HM2$T_FORMAT]);
;  1126          2219  2  DECR J FROM .CLUSTER-1 TO 1 DO
;  1127          2220  2      WRITE_HOMEBLOCK(BUFFER);
;  1128          2221  2  BUFFER[HM2$L_HOMELBN] = .HOMEBLOCK1_LBN;
;  1129          2222  2  DECR J FROM .CLUSTER TO 1 DO
;  1130          2223  2      WRITE_HOMEBLOCK(BUFFER);
;  1131          2224  2  BUFFER[HM2$L_HOMELBN] = .HOMEBLOCK2_LBN;
;  1132          2225  2  DECR J FROM .CLUSTER TO 1 DO
;  1133          2226  2      WRITE_HOMEBLOCK(BUFFER);
;  1134          2227  2
;  1135          2228  2
;  1136          2229  2  ! Finish initializing the VCB, except for the index file window, which is
;  1137          2230  2  ! done after the header is written.
;  1138          2231  2  !
;  1139          2232  2  VCB[VCB_ODS_2] = TRUE;
;  1140          2233  2  VCB[VCB_CLUSTER] = .BUFFER[HM2$W_CLUSTER];
;  1141          2234  2  VCB[VCB_HDR_OFFSET] = .BUFFER[HM2$W_CLUSTER] * 4 + .BUFFER[HM2$W_IBMAPSIZE];
;  1142          2235  2  VCB[VCB_MAXFILIDX] = .BUFFER[HM2$W_IBMAPSIZE] * 4096;
;  1143          2236  2  VCB[VCB_IMAP_LBN] = .IDXFILE_LBN;
;  1144          2237  2
;  1145          2238  2
;  1146          2239  2  ! Initialize the memory resident index file bitmap.  The first block contains
;  1147          2240  2  ! the initially allocated files marked in use; the rest is all zero.
;  1148          2241  2  !
;  1149          2242  2  VCB[VCB_IMAP] = GET_ZERO_VM(.BUFFER[HM2$W_IBMAPSIZE] * 512);
;  1150          2243  2  IF .VCB[VCB_SAVESET]
;  1151          2244  2  THEN .VCB[VCB_IMAP] = %B'111111111'
;  1152          2245  2  ELSE .VCB[VCB_IMAP] = %B'100000111';
;  1153          2246  2
;  1154          2247  2
;  1155          2248  2  ! Construct and write the pending bad block log file header.
;  1156          2249  2  !
;  1157          2250  2  LBN = .IDXFILE_LBN + .BUFFER[HM2$W_IBMAPSIZE] - 1;
;  1158          2251  2  CH$COPY(
;  1159          2252  2      FH2$C_LENGTH+FI2$C_LENGTH, INITIAL_HEADER,
;  1160          2253  2      0,
;  1161          2254  2      512, BUFFER);
;  1162          2255  2  BUFFER[FH2$L_FILEOWNER] = .OUTPUT_ATTBUF[VSR_VOLOWNER];
;  1163          2256  2  BUFFER[FH2$W_FILEPROT] = .OUTPUT_ATTBUF[VSR_FILEPROT];
;  1164          2257  2  BUFFER[FH2$W_RECPROT] = .OUTPUT_ATTBUF[VSR_RECPROT];
;  1165          2258  2  (IDENT_AREA[FI2$Q_CREDATE]) = .(OUTPUT_ATTBUF[VSR_VOLDATE]);
;  1166          2259  2  (IDENT_AREA[FI2$Q_CREDATE]+4) = .(OUTPUT_ATTBUF[VSR_VOLDATE]+4);
;  1167          2260  2  (IDENT_AREA[FI2$Q_REVDATE]) = .(OUTPUT_ATTBUF[VSR_VOLDATE]);
;  1168          2261  2  (IDENT_AREA[FI2$Q_REVDATE]+4) = .(OUTPUT_ATTBUF[VSR_VOLDATE]+4);
;  1169          2262  2  CHECKSUM(BUFFER);
;  1170          2263  2  WRITE_BLOCK(.LBN + FID$C_BADLOG, BUFFER);
;  1171          2264  2
;  1172          2265  2
;  1173          2266  2  ! Construct and write the index file header.
;  1174          2267  2  !
```

```
;1175     2268  2 BUFFER[FH2$W_FID_NUM] = FID$C_INDEXF;
;1176     2269  2 BUFFER[FH2$W_FID_SEQ] = FID$C_INDEXF;
;1177     2270  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_RSIZE] = 512;
;1178     2271  2 BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_MAXREC] = 512;
;1179     2272  2 BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] =
;1180     2273  2     ROT(.CLUSTER*4 ; .IDXFILE_CNT, 16);
;1181     2274  2 BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] =
;1182     2275  2     ROT(.CLUSTER*4 ; (.OUTPUT_ATTBUF[VSR_MAXFILES]+4095)/4096 + .OUTPUT_ATTBUF[VSR_MAXFILNUM] + 1, 16);
;1183     2276  2 BUFFER[FH2$L_HIGHWATER] = .CLUSTER*4 +
;1184     2277  2     (.OUTPUT_ATTBUF[VSR_MAXFILES]+4095)/4096 + .OUTPUT_ATTBUF[VSR_MAXFILNUM] + 1;
;1185     2278  2 CH$MOVE(6, UPLIT BYTE('INDEXF'), IDENT_AREA[FI2$T_FILENAME]);
;1186     2279  2 MAP_COUNT = .BOOTBLOCK_CNT;
;1187     2280  2 MAP_LBN = .BOOTBLOCK_LBN;
;1188     2281  2 INCR J FROM BOOTBLOCK_IDX + 1 TO IDXFILE_IDX DO
;1189     2282  3     BEGIN
;1190     2283  3     IF .MAP_COUNT + .MAP_LBN EQL .ALLOC_TABLE_LBN[.J]
;1191     2284  3     THEN
;1192     2285  3         MAP_COUNT = .MAP_COUNT + .ALLOC_TABLE_CNT[.J]
;1193     2286  3     ELSE
;1194     2287  4         BEGIN
;1195     2288  4         MAKE_POINTER(BUFFER, .MAP_COUNT, .MAP_LBN);
;1196     2289  4         MAP_COUNT = .ALLOC_TABLE_CNT[.J];
;1197     2290  4         MAP_LBN = .ALLOC_TABLE_LBN[.J];
;1198     2291  3         END;
;1199     2292  2     END;
;1200     2293  2 MAKE_POINTER(BUFFER, .MAP_COUNT, .MAP_LBN);
;1201     2294  2 CHECKSUM(BUFFER);
;1202     2295  2 WRITE_BLOCK(.LBN + FID$C_INDEXF, BUFFER);
;1203     2296  2 INCR J FROM 0 TO .CLUSTER-1
;1204     2297  2 DO WRITE_BLOCK(.IDXHDR2_LBN+.J, BUFFER);
;1205     2298  2 CREATE_WINDOW(BUFFER, .VCB[VCB_RVN], VCB[VCB_INDEXF], 1, 0);
;1206     2299  2 BBLOCK [.VCB[VCB_INDEXF], WCB_CUR_HWM] = .CLUSTER*4 +
;1207     2300  2     (.OUTPUT_ATTBUF[VSR_MAXFILES]+4095)/4096 + 9 + 1;
;1208     2301  2
;1209     2302  2
;1210     2303  2 ! Construct and write the bad block file header.
;1211     2304  2 !
;1212     2305  2 CH$FILL(0, 512-FH2$C_LENGTH-FI2$C_LENGTH, BUFFER+FH2$C_LENGTH+FI2$C_LENGTH);
;1213     2306  2 BUFFER[FH2$B_MAP_INUSE] = 0;
;1214     2307  2 BUFFER[FH2$W_FID_NUM] = FID$C_BADBLK;
;1215     2308  2 BUFFER[FH2$W_FID_SEQ] = FID$C_BADBLK;
;1216     2309  2 CH$MOVE(6, UPLIT BYTE('BADBLK'), IDENT_AREA[FI2$T_FILENAME]);
;1217     2310  2 MAP_COUNT = 0;
;1218     2311  2 BAD = OUTPUT_BAD[BAD_DESC];
;1219     2312  2 INCR J FROM 0 TO .OUTPUT_BAD[BAD_NUMDESC]-1 DO
;1220     2313  3     BEGIN
;1221     2314  3     MAP_COUNT = .MAP_COUNT + .BAD[BAD_COUNT];
;1222     2315  3     IF NOT MAKE_POINTER(BUFFER, .BAD[BAD_COUNT], .BAD[BAD_LBN])
;1223     2316  3     THEN SIGNAL(BACKUP$_MAXBAD, 1, VCB[VCB_DEVICE]);
;1224     2317  3     BAD = .BAD + BAD_S_DESC;
;1225     2318  2     END;
;1226     2319  2 BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT(.MAP_COUNT, 16);
;1227     2320  2 BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT(.MAP_COUNT+1, 16);
;1228     2321  2 BUFFER[FH2$L_HIGHWATER] = .MAP_COUNT + 1;
;1229     2322  2 CHECKSUM(BUFFER);
;1230     2323  2 WRITE_BLOCK(.LBN + FID$C_BADBLK, BUFFER);
;1231     2324  2
```

```
;  1232          2325  2
;  1233          2326  2 ! Construct and write the storage map file header.
;  1234          2327  2 !
;  1235          2328  2 CH$FILL(0, 512-FH2$C_LENGTH-FI2$C_LENGTH, BUFFER+FH2$C_LENGTH+FI2$C_LENGTH);
;  1236          2329  2 BUFFER[FH2$B_MAP_INUSE] = 0;
;  1237          2330  2 BUFFER[FH2$W_FID_NUM] = FID$C_BITMAP;
;  1238          2331  2 BUFFER[FH2$W_FID_SEQ] = FID$C_BITMAP;
;  1239          2332  2 BUFFER[FH2$V_CONTIG] = TRUE;
;  1240          2333  2 BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT(.BITMAP_CNT, 16);
;  1241          2334  2 BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT((.VOLUME_SIZE/.CLUSTER+4095)/4096 + 2, 16);
;  1242          2335  2 BUFFER[FH2$L_HIGHWATER] = (.VOLUME_SIZE/.CLUSTER+4095)/4096 + 2;
;  1243          2336  2 CH$MOVE(6, UPLIT BYTE('BITMAP'), IDENT_AREA[FI2$T_FILENAME]);
;  1244          2337  2 MAKE_POINTER(BUFFER, .BITMAP_CNT, .BITMAP_LBN);
;  1245          2338  2 CHECKSUM(BUFFER);
;  1246          2339  2 WRITE_BLOCK(.LBN + FID$C_BITMAP, BUFFER);
;  1247          2340  2
;  1248          2341  2 ! If this is a save set volume we are initializing, also set up the headers
;  1249          2342  2 ! of the miscellaneous reserved files. If the volume is being restored to,
;  1250          2343  2 ! then the restore process will create these files.
;  1251          2344  2 !
;  1252          2345  2
;  1253          2346  2 IF .VCB[VCB_SAVESET]
;  1254          2347  2 THEN
;  1255          2348  3     BEGIN
;  1256          2349  3
;  1257          2350  3 ! Turn the header into the continuation file header and write it.
;  1258          2351  3 !
;  1259          2352  3
;  1260          2353  3     CH$FILL (0, 512-FH2$C_LENGTH-FI2$C_LENGTH, BUFFER+FH2$C_LENGTH+FI2$C_LENGTH);
;  1261          2354  3     BUFFER[FH2$B_MAP_INUSE] = 0;
;  1262          2355  3     BUFFER[FH2$V_CONTIG] = FALSE;
;  1263          2356  3     BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = 0;
;  1264          2357  3     BBLOCK[BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = 1^16;
;  1265          2358  3     BUFFER[FH2$L_HIGHWATER] = 1;
;  1266          2359  3     BUFFER[FH2$W_FID_NUM] = FID$C_CONTIN;
;  1267          2360  3     BUFFER[FH2$W_FID_SEQ] = FID$C_CONTIN;
;  1268          2361  3     CH$MOVE (6, UPLIT BYTE ('CONTIN'), IDENT_AREA[FI2$T_FILENAME]);
;  1269          2362  3     CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  1270          2363  3     WRITE_BLOCK (.LBN + FID$C_CONTIN, BUFFER);
;  1271          2364  3
;  1272          2365  3 ! Turn the file header into the core image file header and write it.
;  1273          2366  3 !
;  1274          2367  3
;  1275          2368  3     BUFFER[FH2$W_FID_NUM] = FID$C_CORIMG;
;  1276          2369  3     BUFFER[FH2$W_FID_SEQ] = FID$C_CORIMG;
;  1277          2370  3     CH$MOVE (6, UPLIT BYTE ('CORIMG'), IDENT_AREA[FI2$T_FILENAME]);
;  1278          2371  3     CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  1279          2372  3     WRITE_BLOCK (.LBN + FID$C_CORIMG, BUFFER);
;  1280          2373  3
;  1281          2374  3 ! Turn the header into the volume set list file header and write it.
;  1282          2375  3 !
;  1283          2376  3
;  1284          2377  3     BUFFER[FH2$W_FID_NUM] = FID$C_VOLSET;
;  1285          2378  3     BUFFER[FH2$W_FID_SEQ] = FID$C_VOLSET;
;  1286          2379  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_RSIZE] = 64;
;  1287          2380  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_MAXREC] = 64;
;  1288          2381  3     CH$MOVE (6, UPLIT BYTE ('VOLSET'), IDENT_AREA[FI2$T_FILENAME]);
```

```
;  1289          2382  3     CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  1290          2383  3     WRITE_BLOCK (.LBN + FID$C_VOLSET, BUFFER);
;  1291          2384  3
;  1292          2385  3 ! Turn the header into the backup journal file header and write it.
;  1293          2386  3 !
;  1294          2387  3
;  1295          2388  3     BUFFER[FH2$W_FID_NUM] = FID$C_BACKUP;
;  1296          2389  3     BUFFER[FH2$W_FID_SEQ] = FID$C_BACKUP;
;  1297          2390  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_RSIZE] = 512;
;  1298          2391  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$W_MAXREC] = 512;
;  1299          2392  3     CH$MOVE (6, UPLIT BYTE ('BACKUP'), IDENT_AREA[FI2$T_FILENAME]);
;  1300          2393  3     CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  1301          2394  3     WRITE_BLOCK (.LBN + FID$C_BACKUP, BUFFER);
;  1302          2395  3
;  1303          2396  3 ! Turn the file header into the MFD header and write it.
;  1304          2397  3 !
;  1305          2398  3
;  1306          2399  3     BUFFER[FH2$W_FID_NUM] = FID$C_MFD;
;  1307          2400  3     BUFFER[FH2$W_FID_SEQ] = FID$C_MFD;
;  1308          2401  3     BUFFER[FH2$V_CONTIG] = TRUE;
;  1309          2402  3     BUFFER[FH2$V_DIRECTORY] = 1;
;  1310          2403  3     BUFFER[FH2$L_HIGHWATER] = 2;
;  1311          2404  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_EFBLK] = ROT (2, 16);
;  1312          2405  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT (.MFD_CNT, 16);
;  1313          2406  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$B_RTYPE] = FAT$C_VARIABLE;
;  1314          2407  3     BBLOCK [BUFFER[FH2$W_RECATTR], FAT$B_RATTRIB] = FAT$M_NOSPAN;
;  1315          2408  3
;  1316          2409  3     CH$MOVE (10, UPLIT BYTE ('000000.DIR'), IDENT_AREA[FI2$T_FILENAME]);
;  1317          2410  3     MAKE_POINTER (BUFFER, .MFD_CNT, .MFD_LBN);
;  1318          2411  3     CHECKSUM2 (BUFFER, $BYTEOFFSET (FH2$W_CHECKSUM));
;  1319          2412  3     WRITE_BLOCK (.LBN + FID$C_MFD, BUFFER);
;  1320          2413  2     END;
;  1321          2414  2
;  1322          2415  1 END;


                                                    0078B          .BLKB   1
                                                 28 0078C P.AAE:   .BYTE   40
                                                 64 0078D          .BYTE   100
                                                 FF 0078E          .BYTE   -1
                                                 FF 0078F          .BYTE   -1
                                               0000 00790          .WORD   0
                                             02 01  00792          .BYTE   1, 2
                                       0000  0009 0009 00794       .WORD   9, 9, 0
                                       0000  0000 0000 0079A       .WORD   0, 0, 0
                                                 01 007A0          .BYTE   1
                                                 00 007A1          .BYTE   0
                                               0010 007A2          .WORD   16
                                   00010000 00000000 007A4         .LONG   0, 65536
                                               0000 007AC          .WORD   0
                                             00 00  007AE          .BYTE   0, 0
                                               0010 007B0          .WORD   16
                                               0000 007B2          .WORD   0
                  0000  0000  0000  0000  0000 0000 007B4          .WORD   0, 0, 0, 0, 0, 0
                                           00000000 007C0          .LONG   0
                                               0000 007C4          .WORD   0
```

```
                                                        00  00  007C6         .BYTE   0, 0
                                                  00000000  007C8         .LONG   0
                                                      0000  007CC         .WORD   0
                                              0000  0004  0004  007CE         .WORD   4, 4, 0
                                                      0000  007D4         .WORD   0
                                                      0000  007D6         .WORD   0
                                                  00000001  007D8         .LONG   1
20  20  20  31  3B  53  59  53  2E  47  4F  4C  44  41  42  007DC         .ASCII  \BADLOG.SYS;1        \
                                        20  20  20  20  20  007EB
                                                      0001  007F0         .WORD   1
00000000  00000000  00000000  00000000  00000000  00000000  007F2         .LONG   0, 0, 0, 0, 0, 0, 0, 0
                                        00000000  00000000  0080A
                                                        20  00812         .ASCII  \ \
                                                        20  00813         .ASCII  \ \
                                                        20  00814         .ASCII  \ \
                                                        20  00815         .ASCII  \ \
                                                        20  00816         .ASCII  \ \
                                                        20  00817         .ASCII  \ \
                                                        20  00818         .ASCII  \ \
                                                        20  00819         .ASCII  \ \
                                                        20  0081A         .ASCII  \ \
                                                        20  0081B         .ASCII  \ \
                                                        20  0081C         .ASCII  \ \
                                                        20  0081D         .ASCII  \ \
                                                        20  0081E         .ASCII  \ \
                                                        20  0081F         .ASCII  \ \
                                                        20  00820         .ASCII  \ \
                                                        20  00821         .ASCII  \ \
                                                        20  00822         .ASCII  \ \
                                                        20  00823         .ASCII  \ \
                                                        20  00824         .ASCII  \ \
                                                        20  00825         .ASCII  \ \
                                                        20  00826         .ASCII  \ \
                                                        20  00827         .ASCII  \ \
                                                        20  00828         .ASCII  \ \
                                                        20  00829         .ASCII  \ \
                                                        20  0082A         .ASCII  \ \
                                                        20  0082B         .ASCII  \ \
                                                        20  0082C         .ASCII  \ \
                                                        20  0082D         .ASCII  \ \
                                                        20  0082E         .ASCII  \ \
                                                        20  0082F         .ASCII  \ \
                                                        20  00830         .ASCII  \ \
                                                        20  00831         .ASCII  \ \
                                                        20  00832         .ASCII  \ \
                                                        20  00833         .ASCII  \ \
                                                        20  00834         .ASCII  \ \
                                                        20  00835         .ASCII  \ \
                                                        20  00836         .ASCII  \ \
                                                        20  00837         .ASCII  \ \
                                                        20  00838         .ASCII  \ \
                                                        20  00839         .ASCII  \ \
                                                        20  0083A         .ASCII  \ \
                                                        20  0083B         .ASCII  \ \
                                                        20  0083C         .ASCII  \ \
                                                        20  0083D         .ASCII  \ \
                                                        20  0083E         .ASCII  \ \
```

```
                                              20  0083F          .ASCII  \ \                                  ;
                                              20  00840          .ASCII  \ \                                  ;
                                              20  00841          .ASCII  \ \                                  ;
                                              20  00842          .ASCII  \ \                                  ;
                                              20  00843          .ASCII  \ \                                  ;
                                              20  00844          .ASCII  \ \                                  ;
                                              20  00845          .ASCII  \ \                                  ;
                                              20  00846          .ASCII  \ \                                  ;
                                              20  00847          .ASCII  \ \                                  ;
                                              20  00848          .ASCII  \ \                                  ;
                                              20  00849          .ASCII  \ \                                  ;
                                              20  0084A          .ASCII  \ \                                  ;
                                              20  0084B          .ASCII  \ \                                  ;
                                              20  0084C          .ASCII  \ \                                  ;
                                              20  0084D          .ASCII  \ \                                  ;
                                              20  0084E          .ASCII  \ \                                  ;
                                              20  0084F          .ASCII  \ \                                  ;
                                              20  00850          .ASCII  \ \                                  ;
                                              20  00851          .ASCII  \ \                                  ;
                                              20  00852          .ASCII  \ \                                  ;
                                              20  00853          .ASCII  \ \                                  ;
   20  20  42  31  31  45  4C  49  46  43  45  44  00854 P.AAF:  .ASCII  \DECFILE11B  \                       ;
                           46  58  45  44  4E  49  00860 P.AAG:  .ASCII  \INDEXF\                             ;
                           4B  4C  42  44  41  42  00866 P.AAH:  .ASCII  \BADBLK\                             ;
                           50  41  4D  54  49  42  0086C P.AAI:  .ASCII  \BITMAP\                             ;
                           4E  49  54  4E  4F  43  00872 P.AAJ:  .ASCII  \CONTIN\                             ;
                           47  4D  49  52  4F  43  00878 P.AAK:  .ASCII  \CORIMG\                             ;
                           54  45  53  4C  4F  56  0087E P.AAL:  .ASCII  \VOLSET\                             ;
                           50  55  4B  43  41  42  00884 P.AAM:  .ASCII  \BACKUP\                             ;
           52  49  44  2E  30  30  30  30  30  30  0088A P.AAN:  .ASCII  \000000.DIR\                         ;

                                                     INITIAL_HEADER=     P.AAE


                                            07FC 00000 INIT_INDEX:
                                                                 .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10     ; 2098
                           5A      F9FB  CF  9E 00002            MOVAB   WRITE_BLOCK, R10                     ;
                           59 00000000'  EF  9E 00007            MOVAB   OUTPUT_ATTBUF+52, R9                 ;
                           5E      FE00  CE  9E 0000E            MOVAB   -512(SP), SP                         ;
                           50        24  A9  3C 00013            MOVZWL  OUTPUT_ATTBUF+88, R0                 ; 2142
                                         0F  13 00017            BEQL    1$                                   ;
  0200  8F        00    28   B9              50  2C 00019            MOVC5   R0, @OUTPUT_ATTBUF+92, #0, #512, BUFFER ; 2145
                                         6E     00021                                                         ;
                     04    AE            01  CE 00022            MNEGL   #1, BUFFER+4                         ; 2150
                                         19  11 00026            BRB     2$                                   ; 2142
                6E  F73F   CF            26  28 00028 1$:        MOVC3   #38, BOOT_PROGRAM, BUFFER            ; 2154
  01DA  8F      00  F75F   CF            28  2C 0002E            MOVC5   #40, BOOT_MESSAGE, #0, #474, (R3)    ;
                                         63     00037                                                         ;
        0C      20    D0   B9        CC  A9  2C 00038            MOVC5   OUTPUT_ATTBUF, @OUTPUT_ATTBUF+4, #32, #12, -; 2163
                                     26  AE     0003F                    BUFFER+38                            ;
                                         5E  DD 00041 2$:        PUSHL   SP                                   ; 2165
                                     3C  AB  DD 00043            PUSHL   60(P$)                               ;
                           6A            02  FB 00046            CALLS   #2, WRITE_BLOCK                      ;
  0200  8F        00       6E            00  2C 00049            MOVC5   #0, (SP), #0, #512, BUFFER           ; 2171
                                         6E     00050                                                         ;
                           50        3C  AB  9E 00051            MOVAB   60(P$), R0                           ; 2172
                6E         60            01  C1 00055            ADDL3   #1, (R0), BUFFER                     ;
```

```
                      04   AE        10   AB   D0 00059          MOVL    16(P$), BUFFER+4                                        ; 2173
                      08   AE        0C   A0   D0 0005E          MOVL    12(R0), BUFFER+8                                        ; 2174
                      0C   AE        0C   A9   B0 00063          MOVW    OUTPUT_ATTBUF+64, BUFFER+12                             ; 2175
                           56        14   AB   D0 00068          MOVL    20(P$), R6                                              ; 2176
                      0E   AE             56   B0 0006C          MOVW    R6, BUFFER+14                                           ;
                      10   AE             02   B0 00070          MOVW    #2, BUFFER+16                                           ; 2177
                51    10   AB        08   A0   C3 00074          SUBL3   8(R0), 16(P$), R1                                       ; 2178
                           52        01 A146   3E 0007A          MOVAW   1(R1)[R6], R2                                           ;
                      12   AE             52   B0 0007F          MOVW    R2, BUFFER+18                                           ;
                52                   56        03   C5 00083    MULL3   #3, R6, R2                                              ; 2179
             14 AE                   52        01   A1 00087    ADDW3   #1, R2, BUFFER+20                                       ;
                51                   56        02   78 0008C    ASHL    #2, R6, R1                                              ; 2180
             16 AE                   51        01   A1 00090    ADDW3   #1, R1, BUFFER+22                                       ;
                      18   AE        10   A0   D0 00095          MOVL    16(R0), BUFFER+24                                       ; 2181
                      1C   AE             69   D0 0009A          MOVL    OUTPUT_ATTBUF+52, BUFFER+28                             ; 2182
                50                   69 00000FFF 8F C1 0009E    ADDL3   #4095, OUTPUT_ATTBUF+52, R0                             ; 2183
                51                   50 00001000 8F C7 000A6    DIVL3   #4096, R0, R1                                           ;
                      20   AE             51   B0 000AE          MOVW    R1, BUFFER+32                                           ;
                      22   AE             09   B0 000B2          MOVW    #9, BUFFER+34                                           ;
                           57        0E   A9   3C 000B6          MOVZWL  OUTPUT_ATTBUF+66, R7                                    ; 2184
                      26   AE             57   B0 000BA          MOVW    R7, BUFFER+38                                           ; 2185
                           01             57   B1 000BE          CMPW    R7, #1                                                  ;
                                          06   12 000C1          BNEQ    3$                                                      ; 2186
                      28   AE       FEC3  C9   9B 000C3          MOVZBW  COM_O_SETCOUNT, BUFFER+40                               ; 2188
                      2A   AE         16  A9   B0 000C9 3$:      MOVW    OUTPUT_ATTBUF+74, BUFFER+42                             ; 2189
                      2C   AE         F4  A9   D0 000CE          MOVL    OUTPUT_ATTBUF+40, BUFFER+44                             ; 2190
                      34   AE         10  A9   D0 000D3          MOVL    OUTPUT_ATTBUF+68, BUFFER+52                             ; 2191
                      38   AE         14  A9   B0 000D8          MOVW    OUTPUT_ATTBUF+72, BUFFER+56                             ; 2193
                      3C   AE         E4  A9   7D 000DD          MOVQ    OUTPUT_ATTBUF+24, BUFFER+60                             ; 2194
                      44   AE         1E  A9   B0 000E2          MOVW    OUTPUT_ATTBUF+82, BUFFER+68                             ; 2196
                      46   AE         18  A9   B0 000E7          MOVW    OUTPUT_ATTBUF+76, BUFFER+70                             ; 2198
                      48   AE         2C  A9   7D 000EC          MOVQ    OUTPUT_ATTBUF+96, BUFFER+72                             ; 2199
                      50   AE         34  A9   7D 000F1          MOVQ    OUTPUT_ATTBUF+104, BUFFER+80                            ; 2201
                           50         B8  A9   D0 000F6          MOVL    OUTPUT_BAD, R0                                          ; 2203
                      C8   AD         04  A0   D0 000FA          MOVL    4(R0), BUFFER+456                                       ;
          0C            20             6E        00   2C 000FF    MOVC5   #0, (SP), #32, #12, BUFFER+460                          ; 2204
                                         CC   AD    00104                                                                       ;
                                              57   D5 00106      TSTL    R7                                                      ; 2205
                                              07   13 00108      BEQL    4$                                                      ;
             CC   AD       FED0  C9                   0C   28 0010A    MOVC3   #12, COM_O_STRUCNAME, BUFFER+460                 ; 2207
          0C            20   D0  B9                CC   A9 2C 00111 4$:    MOVC5   OUTPUT_ATTBUF, @OUTPUT_ATTBUF+4, #32, #12, -    ; 2212
                                                   D8   AD    00118                BUFFER+472                                      ;
          0C            20   D8  B9                D4   A9 2C 0011A    MOVC5   OUTPUT_ATTBUF+8, @OUTPUT_ATTBUF+12, #32, -      ; 2217
                                                   E4   AD    00121                #12, BUFFER+484                                 ;
             F0   AD       FE98  CF                   0C   28 00123    MOVC3   #12, P.AAF, BUFFER+496                          ; 2218
                                 52                   56   D0 0012A    MOVL    R6, J                                           ; 2219
                                                      07   11 0012D    BRB     6$                                              ;
                                                      5E   DD 0012F 5$:       PUSHL   SP                                      ; 2220
                                 0000V CF                  01   FB 00131    CALLS   #1, WRITE_HOMEBLOCK                             ;
                                       F6                  52   F5 00136 6$:       SOBGTR  J, 5$                               ;
                                       6E             40   AB   D0 00139          MOVL    64(P$), BUFFER                      ; 2221
                52                   14   AB                  01   C1 0013D    ADDL3   #1, 20(P$), J                           ; 2222
                                                      07   11 00142    BRB     8$                                              ;
                                                      5E   DD 00144 7$:       PUSHL   SP                                      ; 2223
                                 0000V CF                  01   FB 00146    CALLS   #1, WRITE_HOMEBLOCK                             ;
                                       F6                  52   F5 0014B 8$:       SOBGTR  J, 7$                               ;
                                       6E             44   AB   D0 0014E          MOVL    68(P$), BUFFER                      ; 2224
```

```
                        52      14  AB              01  C1 00152          ADDL3   #1, 20(P$), J                                   ; 2225
                                                    07  11 00157          BRB     10$
                                                    5E  DD 00159 9$:       PUSHL   SP                                              ; 2226
                      0000V     CF                  01  FB 0015B          CALLS   #1, WRITE_HOMEBLOCK
                                F6                  52  F5 00160 10$:      SOBGTR  J, 9$
                                52          18      AB  D0 00163          MOVL    24(P$), R2                                      ; 2232
                                  07        A2      02  88 00167          BISB2   #2, 7(R2)
                                50          0E      AE  3C 0016B          MOVZWL  BUFFER+14, R0                                   ; 2233
                                  04        A2      50  B0 0016F          MOVW    R0, 4(R2)
                                53          20      AE  3C 00173          MOVZWL  BUFFER+32, R3                                   ; 2234
                                51                6340  DE 00177          MOVAL   (R3)[R0], R1
                                  1A        A2      51  B0 0017B          MOVW    R1, 26(R2)
                          1C    A2                  0C  78 0017F          ASHL    #12, R3, 28(R2)                                 ; 2235
                                  14        A2      4C  AB  D0 00184      MOVL    76(P$), 20(R2)                                  ; 2236
                          7E                        09  78 00189          ASHL    #9, R3, -(SP)                                   ; 2242
                             00000000G      00      01  FB 0018D          CALLS   #1, GET_ZERO_VM
                                  10        A2      50  D0 00194          MOVL    R0, 16(R2)
                          08      07        A2      03  E1 00198          BBC     #3, 7(R2), 11$                                  ; 2243
                                  10        B2  01FF    8F  3C 0019D      MOVZWL  #511, @16(R2)                                   ; 2244
                                                    06  11 001A3          BRB     12$
                                  10        B2  0107    8F  3C 001A5 11$:  MOVZWL  #263, @16(R2)                                   ; 2245
                          56                        4C  AB  C1 001AB 12$:  ADDL3   76(P$), R3, R6                                 ; 2250
                                                    56  D7 001B0          DECL    LBN
              0200  8F            00      FD3F    CF      00C8  8F  2C 001B2  MOVC5   #200, INITIAL_HEADER, #0, #512, BUFFER       ; 2251
                                                    6E      001BD
                                  3C        AE      F4  A9  D0 001BE      MOVL    OUTPUT_ATTBUF+40, BUFFER+60                     ; 2255
                                  40        AE      12  A9  B0 001C3      MOVW    OUTPUT_ATTBUF+70, BUFFER+64                     ; 2256
                                  38        AE      14  A9  B0 001C8      MOVW    OUTPUT_ATTBUF+72, BUFFER+56                     ; 2257
                                  66        AE      E4  A9  7D 001CD      MOVQ    OUTPUT_ATTBUF+24, IDENT_AREA+22                 ; 2258
                                  6E        AE      E4  A9  7D 001D2      MOVQ    OUTPUT_ATTBUF+24, IDENT_AREA+30                 ; 2260
                                                    5E  DD 001D7          PUSHL   SP                                              ; 2262
                             00000000G      00          01  FB 001D9      CALLS   #1, CHECKSUM
                                                    5E  DD 001E0          PUSHL   SP                                              ; 2263
                                        09      A6  9F 001E2              PUSHAB  9(LBN)
                                    6A              02  FB 001E5          CALLS   #2, WRITE_BLOCK
                                  08    AE  00010001  8F  D0 001E8        MOVL    #65537, BUFFER+8                                ; 2268
                                  16    AE      0200  8F  B0 001F0        MOVW    #512, BUFFER+22                                 ; 2270
                                  24    AE      0200  8F  B0 001F6        MOVW    #512, BUFFER+36                                 ; 2271
                                50          14      AB  D0 001FC          MOVL    20(P$), R0                                      ; 2273
                                51          30  BB40    DE 00200          MOVAL   @48(P$)[R0], R1
                  18    AE              51          10  9C 00205          ROTL    #16, R1, BUFFER+24
                        51              69  00000FFF  8F  C1 0020A        ADDL3   #4095, OUTPUT_ATTBUF+52, R1                     ; 2275
                                        51  00001000  8F  C6 00212        DIVL2   #4096, R1
                                50              6140  DE 00219          MOVAL   (R1)[R0], R0
                                50          04      A9  C0 0021D          ADDL2   OUTPUT_ATTBUF+56, R0
                                                    50  D6 00221          INCL    R0
                  1C    AE              50          10  9C 00223          ROTL    #16, R0, BUFFER+28
                                  4C    AE          50  D0 00228          MOVL    R0, BUFFER+76                                   ; 2277
                  50    AE      FD9B    CF          06  28 0022C          MOVC3   #6, P.AAG, IDENT_AREA                           ; 2278
                                50                  5B  D0 00233          MOVL    P$, R0                                          ; 2279
                                57          20      A0  D0 00236          MOVL    32(R0), MAP_COUNT
                                53          3C      AB  D0 0023A          MOVL    60(P$), MAP_LBN                                 ; 2280
                                52                  01  D0 0023E          MOVL    #1, J                                           ; 2281
                        50              57                  53  C1 00241 13$:  ADDL3   MAP_LBN, MAP_COUNT, R0                  ; 2283
                                  3C  AB42                  50  D1 00245      CMPL    R0, 60(P$)[J]
                                                            07  12 0024A      BNEQ    14$
                                57          20  AB42        C0 0024C          ADDL2   32(P$)[J], MAP_COUNT                            ; 2285
```

```
                                        18  11 00251       BRB     15$                                     ;
                                        53  DD 00253 14$:  PUSHL   MAP_LBN                                 ; 2288
                                        57  DD 00255       PUSHL   MAP_COUNT                               ;
                                  08    AE  9F 00257       PUSHAB  BUFFER                                  ;
                     00000000G  00      03  FB 0025A       CALLS   #3, MAKE_POINTER                        ;
                                57   20 AB42  D0 00261     MOVL    32(P$)[J], MAP_COUNT                    ; 2289
                                53   3C AB42  D0 00266     MOVL    60(P$)[J], MAP_LBN                      ; 2290
                 D2             52      04  F3 0026B 15$:  AOBLEQ  #4, J, 13$                              ; 2281
                                        53  DD 0026F       PUSHL   MAP_LBN                                 ; 2293
                                        57  DD 00271       PUSHL   MAP_COUNT                               ;
                                  08    AE  9F 00273       PUSHAB  BUFFER                                  ;
                     00000000G  00      03  FB 00276       CALLS   #3, MAKE_POINTER                        ;
                                        5E  DD 0027D       PUSHL   SP                                      ; 2294
                     00000000G  00      01  FB 0027F       CALLS   #1, CHECKSUM                            ;
                                        5E  DD 00286       PUSHL   SP                                      ; 2295
                                  01    A6  9F 00288       PUSHAB  1(LBN)                                  ;
                                6A      02  FB 0028B       CALLS   #2, WRITE_BLOCK                         ;
                                53    14    AB  D0 0028E   MOVL    20(P$), R3                              ; 2296
                                52      01  CE 00292       MNEGL   #1, J                                   ;
                                        09  11 00295       BRB     17$                                     ;
                                        5E  DD 00297 16$:  PUSHL   SP                                      ; 2297
                                  48 BB42   9F 00299       PUSHAB  @72(P$)[J]                              ;
                                6A      02  FB 0029D       CALLS   #2, WRITE_BLOCK                         ;
                 F3             52      53  F2 002A0 17$:  AOBLSS  R3, J, 16$                              ;
                                7E      01  7D 002A4       MOVQ    #1, -(SP)                               ; 2298
                                58    18    AB  D0 002A7   MOVL    24(P$), R8                              ;
                                        58  DD 002AB       PUSHL   R8                                      ;
                                7E    06    A8  9A 002AD   MOVZBL  6(R8), -(SP)                            ;
                                      10    AE  9F 002B1   PUSHAB  BUFFER                                  ;
                     00000000G  00      05  FB 002B4       CALLS   #5, CREATE_WINDOW                       ;
                                50      68  D0 002BB       MOVL    (R8), R0                                ; 2299
                                51    14    AB  D0 002BE   MOVL    20(P$), R1                              ;
                 52             69 00000FFF  8F  C1 002C2  ADDL3   #4095, OUTPUT_ATTBUF+52, R2             ; 2300
                                52 00001000  8F  C6 002CA  DIVL2   #4096, R2                               ;
                           0C   A0      0A A241  DE 002D1  MOVAL   10(R2)[R1], 12(R0)                      ;
  0138    8F     00             6E      00  2C 002D7       MOVC5   #0, (SP), #0, #312, BUFFER+200          ; 2305
                                   00C8     CE    002DE                                                     ;
                                      3A    AE  94 002E1   CLRB    BUFFER+58                               ; 2306
                           08   AE 00030003  8F  D0 002E4  MOVL    #196611, BUFFER+8                       ; 2307
            50   AE      FCE1   CF      06  28 002EC       MOVC3   #6, P.AAH, IDENT_AREA                   ; 2309
                                        57  D4 002F3       CLRL    MAP_COUNT                               ; 2310
                 52        B8   A9      08  C1 002F5       ADDL3   #8, OUTPUT_BAD, BAD                     ; 2311
                                54    B8    B9  D0 002FA   MOVL    @OUTPUT_BAD, R4                         ; 2312
                                53      01  CE 002FE       MNEGL   #1, J                                   ;
                                        2B  11 00301       BRB     20$                                     ;
                                57    04    A2  C0 00303 18$: ADDL2 4(BAD), MAP_COUNT                      ; 2314
                                        62  DD 00307       PUSHL   (BAD)                                   ; 2315
                                  04    A2  DD 00309       PUSHL   4(BAD)                                  ;
                                  08    AE  9F 0030C       PUSHAB  BUFFER                                  ;
                     00000000G  00      03  FB 0030F       CALLS   #3, MAKE_POINTER                        ;
                                12      50  E8 00316       BLBS    R0, 19$                                 ;
                                  20    A8  9F 00319       PUSHAB  32(R8)                                  ; 2316
                                        01  DD 0031C       PUSHL   #1                                      ;
                                 00000000G  8F  DD 0031E   PUSHL   #BACKUP$_MAXBAD                         ;
                     00000000G  00      03  FB 00324       CALLS   #3, LIB$SIGNAL                          ;
                                52      08  C0 0032B 19$:  ADDL2   #8, BAD                                 ; 2317
                 D1             53      54  F2 0032E 20$:  AOBLSS  R4, J, 18$                              ; 2312
```

```
                    18   AE                57          10  9C 00332          ROTL    #16, MAP_COUNT, BUFFER+24                       ; 2319
                                           57          D6 00337          INCL    R7                                              ; 2320
                    1C   AE                57          10  9C 00339          ROTL    #16, R7, BUFFER+28
                                     4C    AE          57  D0 0033E          MOVL    R7, BUFFER+76                                   ; 2321
                                           5E          DD 00342          PUSHL   SP                                              ; 2322
                         00000000G   00                01  FB 00344          CALLS   #1, CHECKSUM
                                           5E          DD 0034B          PUSHL   SP                                              ; 2323
                                        03             A6  9F 0034D          PUSHAB  3(LBN)
                                     6A                02  FB 00350          CALLS   #2, WRITE_BLOCK
0138  8F              00                6E          00  2C 00353          MOVC5   #0, (SP), #0, #312, BUFFER+200                  ; 2328
                                     00C8            CE      0035A
                                       3A            AE  94 0035D          CLRB    BUFFER+58                                       ; 2329
                                  08   AE  00020002   8F  D0 00360          MOVL    #131074, BUFFER+8                               ; 2330
                                  34   AE        80   8F  88 00368          BISB2   #128, BUFFER+52                                 ; 2332
                    18   AE       34   AB           10  9C 0036D          ROTL    #16, 52(P$), BUFFER+24                          ; 2333
                    50        0C   AB        14     AB  C7 00373          DIVL3   20(P$), 12(P$), R0                              ; 2334
                                     50  0FFF        C0  9E 00379          MOVAB   4095(R0), R0
                                     50  00001000    8F  C6 0037E          DIVL2   #4096, R0
                                     50             02  C0 00385          ADDL2   #2, R0
                    1C   AE          50             10  9C 00388          ROTL    #16, R0, BUFFER+28
                                  4C   AE           50  D0 0038D          MOVL    R0, BUFFER+76                                   ; 2335
                    50   AE     FC42   CF           06  28 00391          MOVC3   #6, P.AAI, IDENT_AREA                           ; 2336
                                       50            AB  DD 00398          PUSHL   80(P$)                                          ; 2337
                                       34            AB  DD 0039B          PUSHL   52(P$)
                                       08            AE  9F 0039E          PUSHAB  BUFFER
                         00000000G   00                03  FB 003A1          CALLS   #3, MAKE_POINTER
                                           5E          DD 003A8          PUSHL   SP                                              ; 2338
                         00000000G   00                01  FB 003AA          CALLS   #1, CHECKSUM
                                           5E          DD 003B1          PUSHL   SP                                              ; 2339
                                        02             A6  9F 003B3          PUSHAB  2(LBN)
                                     6A                02  FB 003B6          CALLS   #2, WRITE_BLOCK
                                     50          18    AB  D0 003B9          MOVL    24(P$), R0                                      ; 2346
                         01          07   A0           03  E0 003BD          BBS     #3, 7(R0), 21$
                                                        04  003C2          RET
0138  8F              00                6E          00  2C 003C3 21$:     MOVC5   #0, (SP), #0, #312, BUFFER+200                  ; 2353
                                     00C8            CE      003CA
                                       3A            AE  94 003CD          CLRB    BUFFER+58                                       ; 2354
                                  34   AE        80   8F  8A 003D0          BICB2   #128, BUFFER+52                                 ; 2355
                                       18            AE  D4 003D5          CLRL    BUFFER+24                                       ; 2356
                                  1C   AE  00010000   8F  D0 003D8          MOVL    #65536, BUFFER+28                               ; 2357
                                  4C   AE           01  D0 003E0          MOVL    #1, BUFFER+76                                   ; 2358
                                  08   AE  00070007   8F  D0 003E4          MOVL    #458759, BUFFER+8                               ; 2359
                    50   AE     FBED   CF           06  28 003EC          MOVC3   #6, P.AAJ, IDENT_AREA                           ; 2361
                                     7E  01FE        8F  3C 003F3          MOVZWL  #510, -(SP)                                     ; 2362
                                       04            AE  9F 003F8          PUSHAB  BUFFER
                         00000000G   00                02  FB 003FB          CALLS   #2, CHECKSUM2
                                           5E          DD 00402          PUSHL   SP                                              ; 2363
                                        07             A6  9F 00404          PUSHAB  7(LBN)
                                     6A                02  FB 00407          CALLS   #2, WRITE_BLOCK
                                  08   AE  00050005   8F  D0 0040A          MOVL    #327685, BUFFER+8                               ; 2368
                    50   AE     FBCD   CF           06  28 00412          MOVC3   #6, P.AAK, IDENT_AREA                           ; 2370
                                     7E  01FE        8F  3C 00419          MOVZWL  #510, -(SP)                                     ; 2371
                                       04            AE  9F 0041E          PUSHAB  BUFFER
                         00000000G   00                02  FB 00421          CALLS   #2, CHECKSUM2
                                           5E          DD 00428          PUSHL   SP                                              ; 2372
                                        05             A6  9F 0042A          PUSHAB  5(LBN)
                                     6A                02  FB 0042D          CALLS   #2, WRITE_BLOCK
```

```
                          08   AE 00060006    8F   D0 00430          MOVL    #393222, BUFFER+8                       ; 2377
                          16   AE       40    8F   9B 00438          MOVZBW  #64, BUFFER+22                          ; 2379
                          24   AE       40    8F   9B 0043D          MOVZBW  #64, BUFFER+36                          ; 2380
                50   AE     FBA3   CF          06   28 00442          MOVC3   #6, P.AAL, IDENT_AREA                   ; 2381
                                    7E 01FE    8F   3C 00449          MOVZWL  #510, -(SP)                             ; 2382
                                       04      AE   9F 0044E          PUSHAB  BUFFER                                  ;
                     00000000G      00         02   FB 00451          CALLS   #2, CHECKSUM2                           ;
                                               5E   DD 00458          PUSHL   SP                                      ; 2383
                                       06      A6   9F 0045A          PUSHAB  6(LBN)                                  ;
                                    6A         02   FB 0045D          CALLS   #2, WRITE_BLOCK                         ;
                          08   AE 00080008    8F   D0 00460          MOVL    #524296, BUFFER+8                       ; 2388
                          16   AE     0200    8F   B0 00468          MOVW    #512, BUFFER+22                         ; 2390
                          24   AE     0200    8F   B0 0046E          MOVW    #512, BUFFER+36                         ; 2391
                50   AE     FB77   CF          06   28 00474          MOVC3   #6, P.AAM, IDENT_AREA                   ; 2392
                                    7E 01FE    8F   3C 0047B          MOVZWL  #510, -(SP)                             ; 2393
                                       04      AE   9F 00480          PUSHAB  BUFFER                                  ;
                     00000000G      00         02   FB 00483          CALLS   #2, CHECKSUM2                           ;
                                               5E   DD 0048A          PUSHL   SP                                      ; 2394
                                       08      A6   9F 0048C          PUSHAB  8(LBN)                                  ;
                                    6A         02   FB 0048F          CALLS   #2, WRITE_BLOCK                         ;
                          08   AE 00040004    8F   D0 00492          MOVL    #262148, BUFFER+8                       ; 2399
                          34   AE     2080    8F   A8 0049A          BISW2   #8320, BUFFER+53                        ; 2402
                          4C   AE            02   D0 004A0          MOVL    #2, BUFFER+76                           ; 2403
                          1C   AE 00020000    8F   D0 004A4          MOVL    #131072, BUFFER+28                      ; 2404
                18   AE     38   AB          10   9C 004AC          ROTL    #16, 56(P$), BUFFER+24                  ; 2405
                          14   AE     0802    8F   B0 004B2          MOVW    #2050, BUFFER+20                        ; 2406
                50   AE     FB39   CF          0A   28 004B8          MOVC3   #10, P.AAN, IDENT_AREA                  ; 2409
                                       54      AB   DD 004BF          PUSHL   84(P$)                                  ; 2410
                                       38      AB   DD 004C2          PUSHL   56(P$)                                  ;
                                       08      AE   9F 004C5          PUSHAB  BUFFER                                  ;
                     00000000G      00         03   FB 004C8          CALLS   #3, MAKE_POINTER                        ;
                                    7E 01FE    8F   3C 004CF          MOVZWL  #510, -(SP)                             ; 2411
                                       04      AE   9F 004D4          PUSHAB  BUFFER                                  ;
                     00000000G      00         02   FB 004D7          CALLS   #2, CHECKSUM2                           ;
                                               5E   DD 004DE          PUSHL   SP                                      ; 2412
                                       04      A6   9F 004E0          PUSHAB  4(LBN)                                  ;
                                    6A         02   FB 004E3          CALLS   #2, WRITE_BLOCK                         ;
                                               04   004E6          RET                                             ; 2415

; Routine Size:  1255 bytes,    Routine Base:  CODE + 0894
```

```
;  1324          2416  1 %SBTTL 'WRITE_HOMEBLOCK - write home block to volume'
;  1325          2417  1 ROUTINE WRITE_HOMEBLOCK(BUFFER): L_PS NOVALUE=
;  1326          2418  1
;  1327          2419  1 !++
;  1328          2420  1 !
;  1329          2421  1 ! FUNCTIONAL DESCRIPTION:
;  1330          2422  1 !       This routine computes the checksums in the home block currently
;  1331          2423  1 !       in the buffer, writes it, and then increments the block numbers
;  1332          2424  1 !       in the home block for the next write.
;  1333          2425  1 !
;  1334          2426  1 ! INPUT PARAMETERS:
;  1335          2427  1 !       BUFFER          - Pointer to buffer.
;  1336          2428  1 !
;  1337          2429  1 ! IMPLICIT INPUTS:
;  1338          2430  1 !       NONE
;  1339          2431  1 !
;  1340          2432  1 ! OUTPUT PARAMETERS:
;  1341          2433  1 !       NONE
;  1342          2434  1 !
;  1343          2435  1 ! IMPLICIT OUTPUTS:
;  1344          2436  1 !       NONE
;  1345          2437  1 !
;  1346          2438  1 ! ROUTINE VALUE:
;  1347          2439  1 !       NONE
;  1348          2440  1 !
;  1349          2441  1 ! SIDE EFFECTS:
;  1350          2442  1 !       NONE
;  1351          2443  1 !
;  1352          2444  1 !--
;  1353          2445  1
;  1354          2446  2 BEGIN
;  1355          2447  2 MAP
;  1356          2448  2         BUFFER:         REF BBLOCK;             ! Pointer to buffer
;  1357          2449  2 L_DECL;
;  1358          2450  2
;  1359          2451  2
;  1360          2452  2 ! Compute the two checksums and then write the block.
;  1361          2453  2 !
;  1362          2454  2 CHECKSUM2(.BUFFER, $BYTEOFFSET(HM2$W_CHECKSUM1));
;  1363          2455  2 CHECKSUM2(.BUFFER, $BYTEOFFSET(HM2$W_CHECKSUM2));
;  1364          2456  2 WRITE_BLOCK(.BUFFER[HM2$L_HOMELBN], .BUFFER);
;  1365          2457  2
;  1366          2458  2
;  1367          2459  2 ! Advance the block numbers to those of the next home block.
;  1368          2460  2 !
;  1369          2461  2 BUFFER[HM2$L_HOMELBN] = .BUFFER[HM2$L_HOMELBN] + 1;
;  1370          2462  2 BUFFER[HM2$W_HOMEVBN] = .BUFFER[HM2$W_HOMEVBN] + 1;
;  1371          2463  1 END;


                                000C 00000 WRITE_HOMEBLOCK:
                                                .WORD   Save R2,R3                              ; 2417
                     53 00000000G   00  9E 00002        MOVAB   CHECKSUM2, R3                   :
                                    3A  DD 00009        PUSHL   #58                             ; 2454
```

```
                          52       04     AC  D0 0000B        MOVL    BUFFER, R2                              ;
                                          52  DD 0000F        PUSHL   R2                                      ;
                          63              02  FB 00011        CALLS   #2, CHECKSUM2                           ;
                          7E      01FE    8F  3C 00014        MOVZWL  #510, -(SP)                             ; 2455
                                          52  DD 00019        PUSHL   R2                                      ;
                          63              02  FB 0001B        CALLS   #2, CHECKSUM2                           ;
                                          52  DD 0001E        PUSHL   R2                                      ; 2456
                                          62  DD 00020        PUSHL   (R2)                                    ;
                      F4F3    CF          02  FB 00022        CALLS   #2, WRITE_BLOCK                         ;
                                          62  D6 00027        INCL    (R2)                                    ; 2461
                                  10      A2  B6 00029        INCW    16(R2)                                  ; 2462
                                              04 0002C        RET                                             ; 2463

; Routine Size:  45 bytes,    Routine Base:  CODE + 0D7B
```

```
;  1373          2464  1 %SBTTL 'Initial ODS-2 MFD'
;  1374          2465  1 !+
;  1375          2466  1 !
;  1376          2467  1 ! The following are the records of the initial MFD for structure level 2.
;  1377          2468  1 !
;  1378          2469  1 !-
;  1379          2470  1
;  1380          2471  1 BIND
;  1381          2472  1         INITIAL_MFD = PLIT (
;  1382          2473  1                                         ! MFD (itself)
;  1383          2474  1         WORD (22),                     ! record byte count
;  1384          2475  1         WORD (1),                      ! version limit
;  1385          2476  1         BYTE (0),                      ! flags
;  1386          2477  1         BYTE (10),                     ! name byte count
;  1387          2478  1         BYTE ('000000.DIR'),           ! name string
;  1388          2479  1         WORD (1),                      ! version number
;  1389          2480  1         WORD (4, 4, 0),                ! file ID
;  1390          2481  1
;  1391          2482  1                                         ! backup journal file
;  1392          2483  1         WORD (22),                     ! record byte count
;  1393          2484  1         WORD (1),                      ! version limit
;  1394          2485  1         BYTE (0),                      ! flags
;  1395          2486  1         BYTE (10),                     ! name byte count
;  1396          2487  1         BYTE ('BACKUP.SYS'),           ! name string
;  1397          2488  1         WORD (1),                      ! version number
;  1398          2489  1         WORD (8, 8, 0),                ! file ID
;  1399          2490  1
;  1400          2491  1                                         ! bad block file
;  1401          2492  1         WORD (22),                     ! record byte count
;  1402          2493  1         WORD (1),                      ! version limit
;  1403          2494  1         BYTE (0),                      ! flags
;  1404          2495  1         BYTE (10),                     ! name byte count
;  1405          2496  1         BYTE ('BADBLK.SYS'),           ! name string
;  1406          2497  1         WORD (1),                      ! version number
;  1407          2498  1         WORD (3, 3, 0),                ! file ID
;  1408          2499  1
;  1409          2500  1                                         ! pending bad block log
;  1410          2501  1         WORD (22),                     ! record byte count
;  1411          2502  1         WORD (1),                      ! version limit
;  1412          2503  1         BYTE (0),                      ! flags
;  1413          2504  1         BYTE (10),                     ! name byte count
;  1414          2505  1         BYTE ('BADLOG.SYS'),           ! name string
;  1415          2506  1         WORD (1),                      ! version number
;  1416          2507  1         WORD (9, 9, 0),                ! file ID
;  1417          2508  1
;  1418          2509  1                                         ! storage bitmap file
;  1419          2510  1         WORD (22),                     ! record byte cout
;  1420          2511  1         WORD (1),                      ! version limit
;  1421          2512  1         BYTE (0),                      ! flags
;  1422          2513  1         BYTE (10),                     ! name byte count
;  1423          2514  1         BYTE ('BITMAP.SYS'),           ! name string
;  1424          2515  1         WORD (1),                      ! version .umber
;  1425          2516  1         WORD (2, 2, 0),                ! file ID
;  1426          2517  1
;  1427          2518  1                                         ! standard continuation file
;  1428          2519  1         WORD (22),                     ! record byte count
;  1429          2520  1         WORD (1),                      ! v%rsion limit
```

```
;  1430          2521  1          BYTE (0),                       ! flags
;  1431          2522  1          BYTE (10),                      ! name byte count
;  1432          2523  1          BYTE ('CONTIN.SYS'),            ! name string
;  1433          2524  1          WORD (1),                       ! version number
;  1434          2525  1          WORD (7, 7, 0),                 ! file ID
;  1435          2526  1
;  1436          2527  1                                          ! core image file
;  1437          2528  1          WORD (22),                      ! record byte count
;  1438          2529  1          WORD (1),                       ! version limit
;  1439          2530  1          BYTE (0),                       ! flags
;  1440          2531  1          BYTE (10),                      ! name byte count
;  1441          2532  1          BYTE ('CORIMG.SYS'),            ! name string
;  1442          2533  1          WORD (1),                       ! version number
;  1443          2534  1          WORD (5, 5, 0),                 ! file ID
;  1444          2535  1
;  1445          2536  1                                          ! index file
;  1446          2537  1          WORD (22),                      ! record byte count
;  1447          2538  1          WORD (1),                       ! version limit
;  1448          2539  1          BYTE (0),                       ! flags
;  1449          2540  1          BYTE (10),                      ! name byte count
;  1450          2541  1          BYTE ('INDEXF.SYS'),            ! name string
;  1451          2542  1          WORD (1),                       ! version number
;  1452          2543  1          WORD (1, 1, 0),                 ! file ID
;  1453          2544  1
;  1454          2545  1                                          ! volume set list file
;  1455          2546  1          WORD (22),                      ! record byte count
;  1456          2547  1          WORD (1),                       ! version limit
;  1457          2548  1          BYTE (0),                       ! flags
;  1458          2549  1          BYTE (10),                      ! name byte count
;  1459          2550  1          BYTE ('VOLSET.SYS'),            ! name string
;  1460          2551  1          WORD (1),                       ! version number
;  1461          2552  1          WORD (6, 6, 0),                 ! file ID
;  1462          2553  1
;  1463          2554  1          WORD (-1)                       ! end marker
;  1464          2555  1
;  1465          2556  1          );
```

```
;1467        2557  1 %SBTTL 'INIT_MFD - initialize ODS-2 MFD'
;1468        2558  1 ROUTINE INIT_MFD : L_PS NOVALUE =
;1469        2559  1
;1470        2560  1 !++
;1471        2561  1 !
;1472        2562  1 ! FUNCTIONAL DESCRIPTION:
;1473        2563  1 !
;1474        2564  1 !       This routine writes the initial master file directory.
;1475        2565  1 !
;1476        2566  1 !
;1477        2567  1 ! CALLING SEQUENCE:
;1478        2568  1 !       INIT_MFD ()
;1479        2569  1 !
;1480        2570  1 ! INPUT PARAMETERS:
;1481        2571  1 !       NONE
;1482        2572  1 !
;1483        2573  1 ! IMPLICIT INPUTS:
;1484        2574  1 !       allocation table
;1485        2575  1 !
;1486        2576  1 ! OUTPUT PARAMETERS:
;1487        2577  1 !       NONE
;1488        2578  1 !
;1489        2579  1 ! IMPLICIT OUTPUTS:
;1490        2580  1 !       NONE
;1491        2581  1 !
;1492        2582  1 ! ROUTINE VALUE:
;1493        2583  1 !       NONE
;1494        2584  1 !
;1495        2585  1 ! SIDE EFFECTS:
;1496        2586  1 !       initial MFD written
;1497        2587  1 !
;1498        2588  1 !--
;1499        2589  1
;1500        2590  2 BEGIN
;1501        2591  2
;1502        2592  2 LOCAL
;1503        2593  2         BUFFER          : BBLOCK [512]; ! I/O buffer
;1504        2594  2
;1505        2595  2 L_DECL;
;1506        2596  2
;1507        2597  2
;1508        2598  2 ! Simply copy the MFD records into the buffer, zero filled and write it.
;1509        2599  2 !
;1510        2600  2
;1511        2601  2 CH$COPY (.(INITIAL_MFD-4)*4, INITIAL_MFD,
;1512        2602  2          0, 512, BUFFER);
;1513        2603  2
;1514        2604  2 WRITE_BLOCK (.MFD_LBN, BUFFER);
;1515        2605  2
;1516        2606  1 END;                                     ! end of routine INIT_MFD


                                00000037  00DA8             .LONG   55
                                    0016  00DAC P.AAO:      .WORD   22
                                    0001  00DAE             .WORD   1
                                      00  00DB0             .BYTE   0
```

```
                                                                 0A  00DB1          .BYTE   10
                52  49  44  2E  30  30  30  30  30  30           00DB2          .ASCII  \000000.DIR\
                                                         0001    00DBC          .WORD   1
                                                 0000    0004    00DBE          .WORD   4, 4, 0
                                                         0016    00DC4          .WORD   22
                                                         0001    00DC6          .WORD   1
                                                           00    00DC8          .BYTE   0
                                                           0A    00DC9          .BYTE   10
                53  59  53  2E  50  55  4B  43  41  42           00DCA          .ASCII  \BACKUP.SYS\
                                                         0001    00DD4          .WORD   1
                                                 0000    0008    00DD6          .WORD   8, 8, 0
                                                         0016    00DDC          .WORD   22
                                                         0001    00DDE          .WORD   1
                                                           00    00DE0          .BYTE   0
                                                           0A    00DE1          .BYTE   10
                53  59  53  2E  4B  4C  42  44  41  42           00DE2          .ASCII  \BADBLK.SYS\
                                                         0001    00DEC          .WORD   1
                                                 0000    0003    00DEE          .WORD   3, 3, 0
                                                         0016    00DF4          .WORD   22
                                                         0001    00DF6          .WORD   1
                                                           00    00DF8          .BYTE   0
                                                           0A    00DF9          .BYTE   10
                53  59  53  2E  47  4F  4C  44  41  42           00DFA          .ASCII  \BADLOG.SYS\
                                                         0001    00E04          .WORD   1
                                                 0000    0009    00E06          .WORD   9, 9, 0
                                                         0016    00E0C          .WORD   22
                                                         0001    00E0E          .WORD   1
                                                           00    00E10          .BYTE   0
                                                           0A    00E11          .BYTE   10
                53  59  53  2E  50  41  4D  54  49  42           00E12          .ASCII  \BITMAP.SYS\
                                                         0001    00E1C          .WORD   1
                                                 0000    0002    00E1E          .WORD   2, 2, 0
                                                         0016    00E24          .WORD   22
                                                         0001    00E26          .WORD   1
                                                           00    00E28          .BYTE   0
                                                           0A    00E29          .BYTE   10
                53  59  53  2E  4E  49  54  4E  4F  43           00E2A          .ASCII  \CONTIN.SYS\
                                                         0001    00E34          .WORD   1
                                                 0000    0007    00E36          .WORD   7, 7, 0
                                                         0016    00E3C          .WORD   22
                                                         0001    00E3E          .WORD   1
                                                           00    00E40          .BYTE   0
                                                           0A    00E41          .BYTE   10
                53  59  53  2E  47  4D  49  52  4F  43           00E42          .ASCII  \CORIMG.SYS\
                                                         0001    00E4C          .WORD   1
                                                 0000    0005    00E4E          .WORD   5, 5, 0
                                                         0016    00E54          .WORD   22
                                                         0001    00E56          .WORD   1
                                                           00    00E58          .BYTE   0
                                                           0A    00E59          .BYTE   10
                53  59  53  2E  46  58  45  44  4E  49           00E5A          .ASCII  \INDEXF.SYS\
                                                         0001    00E64          .WORD   1
                                                 0000    0001    00E66          .WORD   1, 1, 0
                                                         0016    00E6C          .WORD   22
                                                         0001    00E6E          .WORD   1
                                                           00    00E70          .BYTE   0
                                                           0A    00E71          .BYTE   10
```

```
                    53  59  53  2E  54  45  53  4C  4F  56  00E72         .ASCII  \VOLSET.SYS\                    ;
                                                        0001  00E7C         .WORD   1                               ;
                                                0000    0006  0006  00E7E         .WORD   6, 6, 0                 ;
                                                              FFFF  00E84         .WORD   -1                      ;
                                                                    00E86         .BLKB   2                       ;

                                                                          INITIAL_MFD=      P.AAO


                                                                    003C 00000 INIT_MFD:
                                                                          .WORD   Save R2,R3,R4,R5                ; 2558
                                                    5E      FE00    CE  9E 00002        MOVAB   -512(SP), SP            ;
                                  50        FF14    CF            02  78 00007        ASHL    #2, INITIAL_MFD-4, R0   ; 2601
    0200    8F                    00        FF12    CF            50  2C 0000D        MOVC5   R0, INITIAL_MFD, #0, #512, BUFFER
                                                                    6E    00016
                                                                    5E  DD 00017        PUSHL   SP                      ; 2604
                                                          54        AB  DD 00019        PUSHL   84(P$)                  ;
                                                FF3EC   CF            02  FB 0001C        CALLS   #2, WRITE_BLOCK         ;
                                                                    04 00021        RET                             ; 2606

; Routine Size:  34 bytes,    Routine Base:  CODE + 0E88
```

```
;  1518          2607  1 %SBTTL 'INITIALIZE_VOLUME - main volume initialization routine'
;  1519          2608  1 GLOBAL ROUTINE INITIALIZE_VOLUME(PVCB,PDEVCHAR): NOVALUE=
;  1520          2609  1
;  1521          2610  1 !++
;  1522          2611  1 !
;  1523          2612  1 ! FUNCTIONAL DESCRIPTION:
;  1524          2613  1 !       This routine initializes an output disk volume during an image restore.
;  1525          2614  1 !
;  1526          2615  1 ! INPUT PARAMETERS:
;  1527          2616  1 !       PVCB            - Pointer to VCB for output volume.
;  1528          2617  1 !       PDEVCHAR        - Pointer to device characteristics for output volume.
;  1529          2618  1 !
;  1530          2619  1 ! IMPLICIT INPUTS:
;  1531          2620  1 !       OUTPUT_ATTBUF   - Contains volume summary attributes.
;  1532          2621  1 !       VCB[VCB_CHAN]   - Contains channel number assigned to volume.
;  1533          2622  1 !
;  1534          2623  1 ! OUTPUT PARAMETERS:
;  1535          2624  1 !       NONE
;  1536          2625  1 !
;  1537          2626  1 ! IMPLICIT OUTPUTS:
;  1538          2627  1 !       NONE
;  1539          2628  1 !
;  1540          2629  1 ! ROUTINE VALUE:
;  1541          2630  1 !       NONE
;  1542          2631  1 !
;  1543          2632  1 ! SIDE EFFECTS:
;  1544          2633  1 !       NONE
;  1545          2634  1 !
;  1546          2635  1 !--
;  1547          2636  1
;  1548          2637  2 BEGIN
;  1549          2638  2 LOCAL
;  1550          2639  2         P$AREA:         VECTOR[P$SIZE],         ! Impure area
;  1551          2640  2         ACB:            REF BBLOCK,             ! Pointer to ACB
;  1552          2641  2         C;                                      ! Temp
;  1553          2642  2 GLOBAL REGISTER
;  1554          2643  2         P$ = 11:        REF VECTOR;             ! Impure area base register
;  1555          2644  2
;  1556          2645  2
;  1557          2646  2 ! Initialize.
;  1558          2647  2 !
;  1559          2648  2 P$ = P$AREA;
;  1560          2649  2 CH$FILL(0, 2*ALLOC_MAX*4, _ALLOC_TABLE_CNT);
;  1561          2650  2 _VCB = .PVCB;
;  1562          2651  2 _DEVCHAR = .PDEVCHAR;
;  1563          2652  2 CLUSTER = .OUTPUT_ATTBUF[VSR_CLUSTER];
;  1564          2653  2 STRUCLEV_1 = (.OUTPUT_ATTBUF[VSR_STRUCLEV] EQL 1);
;  1565          2654  2
;  1566          2655  2
;  1567          2656  2 ! Strip trailing spaces from the volume name.  This is necessary so that it can
;  1568          2657  2 ! be zero-filled for an ODS-1 header.
;  1569          2658  2 !
;  1570          2659  2 DECR J FROM .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH]-1 TO 0 DO
;  1571          2660  3     BEGIN
;  1572          2661  3     IF .VECTOR[.BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER], .J ;,BYTE] EQL %C' '
;  1573          2662  3     THEN
;  1574          2663  3         BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH] =
```

```
;  1575          2664  3              .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH] - 1;
;  1576          2665  2      END;
;  1577          2666  2
;  1578          2667  2
;  1579          2668  2  ! Get the channel number assigned to the device.
;  1580          2669  2  !
;  1581          2670  2  CHANNEL = .VCB[VCB_CHAN];
;  1582          2671  2
;  1583          2672  2
;  1584          2673  2  ! Get the bad block information.
;  1585          2674  2  !
;  1586          2675  2  OUTPUT_BAD = GET_BADBLOCKS(.VCB[VCB_FAB], .CHANNEL, ._DEVCHAR, .CLUSTER);
;  1587          2676  2
;  1588          2677  2
;  1589          2678  2  ! Compute volume size, rounded up to next cluster boundary.
;  1590          2679  2  !
;  1591          2680  2  VOLUME_SIZE = (.DEVCHAR[DIB$L_MAXBLOCK] + .CLUSTER - 1) / .CLUSTER * .CLUSTER;
;  1592          2681  2
;  1593          2682  2
;  1594          2683  2  ! Allocate ACB to describe entire volume.
;  1595          2684  2  !
;  1596          2685  2  ACB = GET_VM(ACB_S_ENTRY);
;  1597          2686  2  VCB[VCB_ACB_FLINK] = .ACB;
;  1598          2687  2  VCB[VCB_ACB_BLINK] = .ACB;
;  1599          2688  2  ACB[ACB_FLINK] = VCB[VCB_ACB_FLINK];
;  1600          2689  2  ACB[ACB_BLINK] = VCB[VCB_ACB_FLINK];
;  1601          2690  2  ACB[ACB_COUNT] = .VOLUME_SIZE;
;  1602          2691  2  ACB[ACB_LBN] = 0;
;  1603          2692  2
;  1604          2693  2
;  1605          2694  2  ! Unless the user has preinitialized the volume, increase the maximum number
;  1606          2695  2  ! of files if warranted by the output volume size.
;  1607          2696  2  !
;  1608          2697  2  IF
;  1609          2698  2      .QUAL[QUAL_INIT] AND
;  1610          2699  2      .OUTPUT_ATTBUF[VSR_MAXFILES] LSSU
;  1611          2700  3          .DEVCHAR[DIB$L_MAXBLOCK] / ((.CLUSTER + 1) * 2)
;  1612          2701  2  THEN
;  1613          2702  2      OUTPUT_ATTBUF[VSR_MAXFILES] =
;  1614          2703  2          .DEVCHAR[DIB$L_MAXBLOCK] / ((.CLUSTER + 1) * 2);
;  1615          2704  2
;  1616          2705  2
;  1617          2706  2  ! Now verify the parameters against the volume size and characteristics.
;  1618          2707  2  !
;  1619          2708  2  IF .STRUCLEV_1
;  1620          2709  2  THEN
;  1621          2710  3      BEGIN
;  1622          2711  3      IF .DEVCHAR[DIB$L_MAXBLOCK] GTRU 255^12
;  1623          2712  3      THEN
;  1624          2713  3          SIGNAL(BACKUP$_LARGECNT, 1, VCB[VCB_DEVICE]);
;  1625          2714  3      IF .OUTPUT_ATTBUF[VSR_MAXFILES] GTRU 65500
;  1626          2715  3      THEN OUTPUT_ATTBUF[VSR_MAXFILES] = 65500;
;  1627          2716  2      END;
;  1628          2717  2
;  1629          2718  2
;  1630          2719  2  ! Check the cluster factor against its lower bound such that the storage map
;  1631          2720  2  ! does not exceed 255 blocks.  Also check for a reasonable minimum number of
```

```
; 1632          2721  2 ! clusters.
; 1633          2722  2 !
; 1634          2723  2 IF
; 1635          2724  2     .VOLUME_SIZE / .CLUSTER GTRU 255^12 OR
; 1636          2725  2     .VOLUME_SIZE / .CLUSTER LSS 50
; 1637          2726  2 THEN
; 1638          2727  2     SIGNAL(BACKUP$_CLUSTER, 1, VCB[VCB_DEVICE]);
; 1639          2728  2
; 1640          2729  2
; 1641          2730  2 ! Increase maximum number of files if necessary to be input index file size.
; 1642          2731  2 !
; 1643          2732  2 IF .OUTPUT_ATTBUF[VSR_MAXFILNUM] GTRU .OUTPUT_ATTBUF[VSR_MAXFILES]
; 1644          2733  2 THEN
; 1645          2734  2     OUTPUT_ATTBUF[VSR_MAXFILES] = .OUTPUT_ATTBUF[VSR_MAXFILNUM];
; 1646          2735  2
; 1647          2736  2
; 1648          2737  2 ! Check maximum number of files against number of clusters.
; 1649          2738  2 !
; 1650          2739  2 C = .VOLUME_SIZE / (.CLUSTER+1);
; 1651          2740  2 IF .OUTPUT_ATTBUF[VSR_MAXFILNUM] GTR .C
; 1652          2741  2 THEN
; 1653          2742  2     SIGNAL(BACKUP$_CLUSTER, 1, VCB[VCB_DEVICE]);
; 1654          2743  2
; 1655          2744  2 IF .OUTPUT_ATTBUF[VSR_MAXFILES] GTR .C
; 1656          2745  2 THEN
; 1657          2746  2     OUTPUT_ATTBUF[VSR_MAXFILES] = .C;
; 1658          2747  2
; 1659          2748  2
; 1660          2749  2 ! Establish the position of the index file.  If /INITIALIZE is specified, or
; 1661          2750  2 ! the original volume is not the same size, place it in the middle.  Otherwise,
; 1662          2751  2 ! propagate the original position.
; 1663          2752  2 !
; 1664          2753  2 ! For middle placement, apply a bias factor equal to the size of a default
; 1665          2754  2 ! initial MFD so that in the usual case the file structure will be allocated
; 1666          2755  2 ! as INIT would allocate it.
; 1667          2756  2 !
; 1668          2757  2 ! For propagated placement, subtract out the size of the storage bitmap to
; 1669          2758  2 ! place the index file bitmap on the same LBN.
; 1670          2759  2 !
; 1671          2760  2 IF
; 1672          2761  2     .OUTPUT_ATTBUF[VSR_INDEXLBN] EQL 0 OR
; 1673          2762  2     .OUTPUT_ATTBUF[VSR_VOLSIZE] NEQ .DEVCHAR[DIB$L_MAXBLOCK]
; 1674          2763  2 THEN
; 1675          2764  3     BEGIN
; 1676          2765  3     OUTPUT_ATTBUF[VSR_INDEXLBN] =
; 1677          2766  3         .DEVCHAR[DIB$L_MAXBLOCK] / 2 + MAXU(2, .CLUSTER);
; 1678          2767  3     END
; 1679          2768  2 ELSE
; 1680          2769  3     BEGIN
; 1681          2770  3     LOCAL
; 1682          2771  3         T;
; 1683          2772  3
; 1684          2773  3     T = (((.VOLUME_SIZE/.CLUSTER + 4095) / 4096) + .CLUSTER) / .CLUSTER * .CLUSTER;
; 1685          2774  3     IF .T LEQU .OUTPUT_ATTBUF[VSR_INDEXLBN]
; 1686          2775  3         THEN OUTPUT_ATTBUF[VSR_INDEXLBN] = .OUTPUT_ATTBUF[VSR_INDEXLBN] - .T
; 1687          2776  3         ELSE OUTPUT_ATTBUF[VSR_INDEXLBN] = 0;
; 1688          2777  2     END;
```

```
2
2
2 ! Now call the routines that do the work of initializing.
2 !
2 INIT_ALLOCATE();
2 INIT_BITMAP();
2 IF .STRUCLEV_1 THEN INIT_INDEX1() ELSE INIT_INDEX();
2 IF .VCB[VCB_SAVESET] THEN INIT_MFD();
2
2
2 ! Note that the initialization is complete.  Record the volume label.
2 !
2 VCB[VCB_INIT_HDRS] = ..VCB[VCB_IMAP];
2 VCB[VCB_INIT_DONE] = TRUE;
2 CH$COPY(
2     .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH],
2     .BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER],
2     %C' ',
2     HM2$S_VOLNAME,
2     VCB[VCB_VOLNAME]);
2
2
2 ! Release the bad block descriptor.
2 !
2 FREE_VM(BAD_S_HEADER + .OUTPUT_BAD[BAD_NUMDESC] * BAD_S_DESC, .OUTPUT_BAD);
2 OUTPUT_BAD = 0;
1 END;
```

```
                                        0BFC 00000        .ENTRY  INITIALIZE_VOLUME, Save R2,R3,R4,R5,R6,R7,- ; 2608
                                                                  R8,R9,R11                                  ;
                      59 00000000G  8F  D0 00002         MOVL    #BACKUP$_CLUSTER, R9                        ;
                      58 00000000G  00  9E 00009         MOVAB   LIB$SIGNAL, R8                              ;
                      57 00000000'  EF  9E 00010         MOVAB   OUTPUT_ATTBUF+52, R7                        ;
                      5E        A8  AE  9E 00017         MOVAB   -88(SP), SP                                 ;
                      5B            6E  9E 0001B         MOVAB   P$AREA, P$                                  ; 2648
 38          00       6E            00  2C 0001E         MOVC5   #0, (SP), #0, #56, 32(P$)                   ; 2649
                                20  AB     00023                                                              ;
                 18   AB        04  AC  D0 00025         MOVL    PVCB, 24(P$)                                ; 2650
                      56        1C  AB  9E 0002A         MOVAB   28(P$), R6                                  ;
                      66        08  AC  D0 0002E         MOVL    PDEVCHAR, (R6)                              ; 2651
                      54        14  AB  9E 00032         MOVAB   20(P$), R4                                  ;
                      64        1A  A7  3C 00036         MOVZWL  OUTPUT_ATTBUF+78, (R4)                      ; 2652
                                    50  D4 0003A         CLRL    R0                                          ; 2653
                      01        0D  A7  91 0003C         CMPB    OUTPUT_ATTBUF+65, #1                        ;
                                    02  12 00040         BNEQ    1$                                          ;
                                    50  D6 00042         INCL    R0                                          ;
                 04   AB            50  D0 00044 1$:     MOVL    R0, 4(P$)                                   ;
                      50        CC  A7  3C 00048         MOVZWL  OUTPUT_ATTBUF, J                            ; 2659
                                    0A  11 0004C         BRB     3$                                          ;
                      20   D0 B740  91 0004E 2$:         CMPB    @OUTPUT_ATTBUF+4[J], #32                    ; 2661
                                    03  12 00053         BNEQ    3$                                          ;
                                CC  A7  B7 00055         DECW    OUTPUT_ATTBUF                               ; 2664
                      F3            50  F4 00058 3$:     SOBGEQ  J, 2$                                       ; 2659
```

```
                                50        18    AB  D0 0005B          MOVL    24(P$), R0                                      ; 2670
                                6B        08    A0  3C 0005F          MOVZWL  8(R0), (P$)
                                                64  DD 00063          PUSHL   (R4)                                            ; 2675
                                52              66  D0 00065          MOVL    (R6), R2
                                                52  DD 00068          PUSHL   R2
                                                6B  DD 0006A          PUSHL   (P$)
                                50        18    AB  D0 0006C          MOVL    24(P$), R0
                                          30    A0  DD 00070          PUSHL   48(R0)
                          00000000G       00        04  FB 00073      CALLS   #4, GET_BADBLOCKS
                                   B8     A7        50  D0 0007A      MOVL    R0, OUTPUT_BAD
                             52    70     A2        64  C1 0007E      ADDL3   (R4), 112(R2), R2                               ; 2680
                                                     52  D7 00083      DECL    R2
                                52                   64  C6 00085      DIVL2   (R4), R2
                          0C    AB                   52  C5 00088      MULL3   (R4), R2, 12(P$)
                                                     10  DD 0008D      PUSHL   #16                                             ; 2685
                          00000000G       00         01  FB 0008F      CALLS   #1, GET_VM
                                52        18    AB  D0 00096          MOVL    24(P$), R2                                      ; 2686
                                   28     A2        50  D0 0009A      MOVL    ACB, 40(R2)
                                   2C     A2        50  D0 0009E      MOVL    ACB, 44(R2)                                     ; 2687
                                60        28    A2  9E 000A2          MOVAB   40(R2), (ACB)                                   ; 2688
                                   04     A0    28  A2  9E 000A6      MOVAB   40(R2), 4(ACB)                                  ; 2689
                                   08     A0    0C  AB  D0 000AB      MOVL    12(P$), 8(ACB)                                  ; 2690
                                          0C    A0  D4 000B0          CLRL    12(ACB)                                         ; 2691
                             27      FE4E  C7        05  E1 000B3      BBC     #5, QUAL+10, 4$                                 ; 2698
                                51              66  D0 000B9          MOVL    (R6), R1                                        ; 2700
                                50              64  D0 000BC          MOVL    (R4), R0
                                50              02  C4 000BF          MULL2   #2, R0
                                50              02  C0 000C2          ADDL2   #2, R0
                             51       70     A1        50  C7 000C5      DIVL3   R0, 112(R1), R1
                                51              67  D1 000CA          CMPL    OUTPUT_ATTBUF+52, R1
                                                11  1E 000CD          BGEQU   4$
                                51              66  D0 000CF          MOVL    (R6), R1                                        ; 2703
                                50              64  D0 000D2          MOVL    (R4), R0
                                50              02  C4 000D5          MULL2   #2, R0
                                50              02  C0 000D8          ADDL2   #2, R0
                         67          70     A1        50  C7 000DB      DIVL3   R0, 112(R1), OUTPUT_ATTBUF+52
                                29              04  AB  E9 000E0 4$:    BLBC    4(P$), 6$                                       ; 2708
                                50              66  D0 000E4          MOVL    (R6), R0                                        ; 2711
                          000FF000      8F        70  A0  D1 000E7      CMPL    112(R0), #1044480
                                                     0E  1B 000EF      BLEQU   5$
                                                20   A2  9F 000F1      PUSHAB  32(R2)                                          ; 2713
                                                     01  DD 000F4      PUSHL   #1
                                   00000000G         8F  DD 000F6      PUSHL   #BACKUP$_LARGECNT
                                68                   03  FB 000FC      CALLS   #3, LIB$SIGNAL
                          0000FFDC      8F                 67  D1 000FF 5$:    CMPL    OUTPUT_ATTBUF+52, #65500                ; 2714
                                                           05  1B 00106      BLEQU   6$
                                67         FFDC    8F      3C 00108      MOVZWL  #65500, OUTPUT_ATTBUF+52                        ; 2715
                             55       0C     AB              64  C7 0010D 6$:    DIVL3   (R4), 12(P$), R5                        ; 2724
                          000FF000      8F                 55  D1 00112      CMPL    R5, #1044480
                                                           05  1A 00119      BGTRU   7$
                                32                         55  D1 0011B      CMPL    R5, #50                                 ; 2725
                                                           0A  18 0011E      BGEQ    8$
                                                20         A2  9F 00120 7$:    PUSHAB  32(R2)                                  ; 2727
                                                           01  DD 00123      PUSHL   #1
                                                           59  DD 00125      PUSHL   R9
                                68                         03  FB 00127      CALLS   #3, LIB$SIGNAL
                                67        04               A7  D1 0012A 8$:    CMPL    OUTPUT_ATTBUF+56, OUTPUT_ATTBUF+52      ; 2732
```

```
                                    04   1B 0012E       BLEQU   9$                                              ;
                       67      04   A7   D0 00130       MOVL    OUTPUT_ATTBUF+56, OUTPUT_ATTBUF+52              ; 2734
           50          64           01   C1 00134 9$:   ADDL3   #1, (R4), R0                                    ; 2739
           53      0C  AB           50   C7 00138       DIVL3   R0, 12(P$), C
                       53      04   A7   D1 0013D       CMPL    OUTPUT_ATTBUF+56, C                             ; 2740
                                    0A   15 00141       BLEQ    10$
                               20   A2   9F 00143       PUSHAB  32(R2)                                          ; 2742
                                    01   DD 00146       PUSHL   #1
                                    59   DD 00148       PUSHL   R9
                       68           03   FB 0014A       CALLS   #3, LIB$SIGNAL
                       53           67   D1 0014D 10$:  CMPL    OUTPUT_ATTBUF+52, C                             ; 2744
                                    03   15 00150       BLEQ    11$
                       67           53   D0 00152       MOVL    C, OUTPUT_ATTBUF+52                             ; 2746
                       52      20   A7   D0 00155 11$:  MOVL    OUTPUT_ATTBUF+84, R2                            ; 2761
                                    0A   13 00159       BEQL    12$
                       50           66   D0 0015B       MOVL    (R6), R0                                        ; 2762
                   70  A0      F8   A7   D1 0015E       CMPL    OUTPUT_ATTBUF+44, 112(R0)
                                    1A   13 00163       BEQL    14$
                       50           66   D0 00165 12$:  MOVL    (R6), R0                                        ; 2766
           50      70  A0           02   C7 00168       DIVL3   #2, 112(R0), R0
                       51           64   D0 0016D       MOVL    (R4), R1
                       02           51   D1 00170       CMPL    R1, #2
                                    03   1E 00173       BGEQU   13$
                       51           02   D0 00175       MOVL    #2, R1
      20   A7          50           51   C1 00178 13$:  ADDL3   R1, R0, OUTPUT_ATTBUF+84
                                    23   11 0017D       BRB     16$                                             ; 2760
                       50    0FFF   C5   9E 0017F 14$:  MOVAB   4095(R5), R0                                    ; 2773
                       50 00001000  8F   C6 00184       DIVL2   #4096, R0
                       50           64   C0 0018B       ADDL2   (R4), R0
                       50           64   C6 0018E       DIVL2   (R4), R0
                       50           64   C4 00191       MULL2   (R4), T
                       52           50   D1 00194       CMPL    T, R2                                           ; 2774
                                    06   1A 00197       BGTRU   15$
                   20  A7           50   C2 00199       SUBL2   T, OUTPUT_ATTBUF+84                             ; 2775
                                    03   11 0019D       BRB     16$
                               20   A7   D4 0019F 15$:  CLRL    OUTPUT_ATTBUF+84                                ; 2776
                 F131  CF           00   FB 001A2 16$:  CALLS   #0, INIT_ALLOCATE                               ; 2782
                 F285  CF           00   FB 001A7       CALLS   #0, INIT_BITMAP                                 ; 2783
                       07      04   AB   E9 001AC       BLBC    4(P$), 17$                                      ; 2784
                 F3AB  CF           00   FB 001B0       CALLS   #0, INIT_INDEX1
                                    05   11 001B5       BRB     18$
                 F82E  CF           00   FB 001B7 17$:  CALLS   #0, INIT_INDEX
                       50      18   AB   D0 001BC 18$:  MOVL    24(P$), R0                                      ; 2785
           05      07  A0           03   E1 001C0       BBC     #3, 7(R0), 19$
                 FE14  CF           00   FB 001C5       CALLS   #0, INIT_MFD
                       50      18   AB   D0 001CA 19$:  MOVL    24(P$), R0                                      ; 2790
                   18  A0      10   B0   B0 001CE       MOVW    @16(R0), 24(R0)
                   07  A0           04   88 001D3       BISB2   #4, 7(R0)                                       ; 2791
 0C        20      D0  B7      CC   A7   2C 001D7       MOVC5   OUTPUT_ATTBUF, @OUTPUT_ATTBUF+4, #32, #12, -    ; 2797
                               38   A0      001DE               56(R0)
                               B8   A7   DD 001E0       PUSHL   OUTPUT_BAD                                      ; 2802
                       50      B8   B7   D0 001E3       MOVL    @OUTPUT_BAD, R0
           7E          50           03   78 001E7       ASHL    #3, R0, -(SP)
                       6E           08   C0 001EB       ADDL2   #8, (SP)
            00000000G  00           02   FB 001EE       CALLS   #2, FREE_VM
                               B8   A7   D4 001F5       CLRL    OUTPUT_BAD                                      ; 2803
                                         04 001F8       RET                                                     ; 2804
```

; Routine Size: 505 bytes, Routine Base: CODE + 0EAA

```
; 1717          2805  1 END
; 1718          2806  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  COMMON                         2124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                           4259  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       129         0      1000          00:02.1


;                                   COMMAND QUALIFIERS

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

; Size:          3576 code + 2807 data bytes
; Run Time:         01:18.0
; Elapsed Time:     04:09.3
; Lines/CPU Min:     2157
; Lexemes/CPU-Min: 32869
; Memory Used:  524 pages
; Compilation Complete
```

TAPEUTIL
LIS

STAINVOL
LIS

UTILITY
LIS

STATOSYS
LIS

STAPARSE
LIS