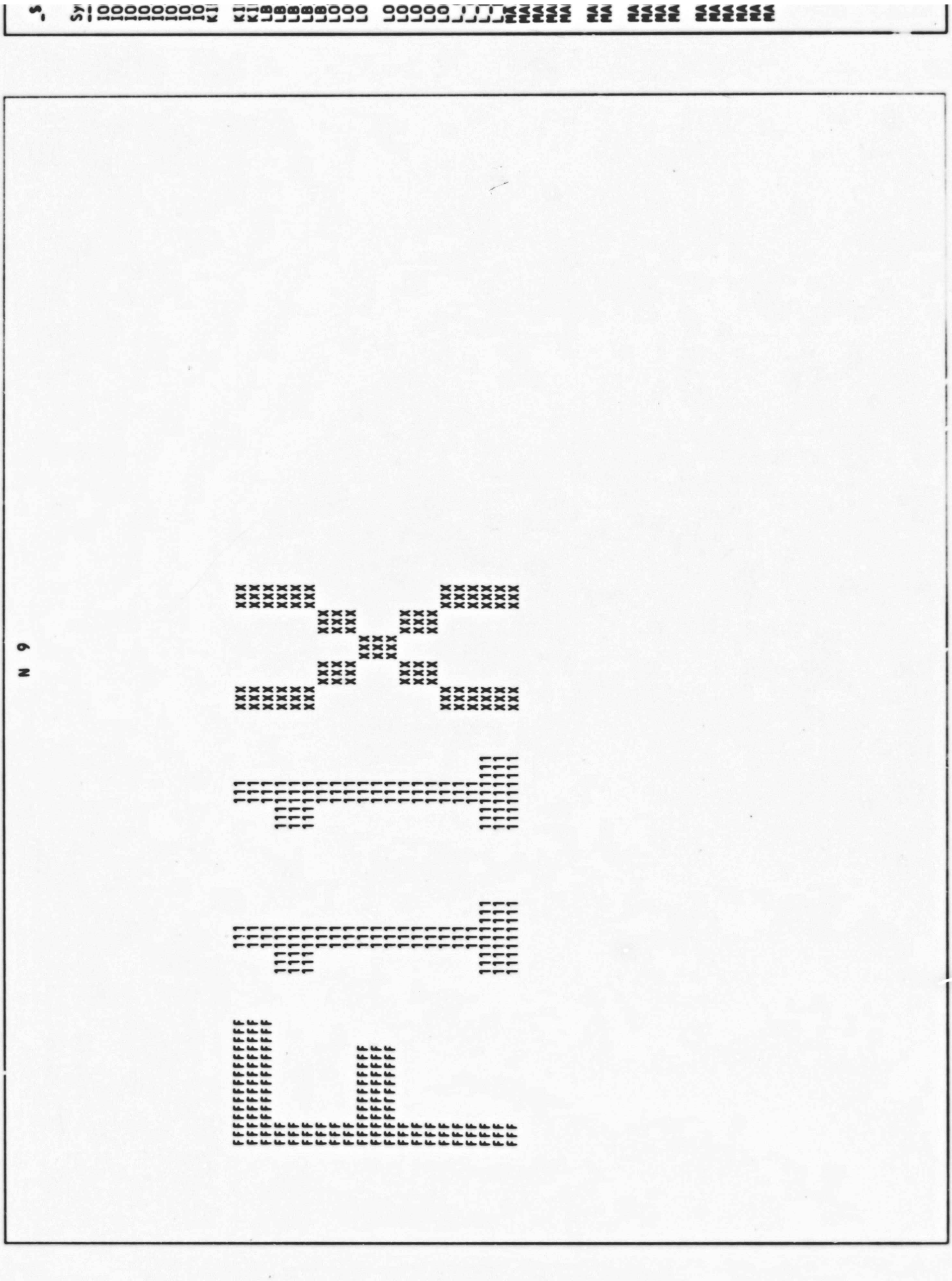

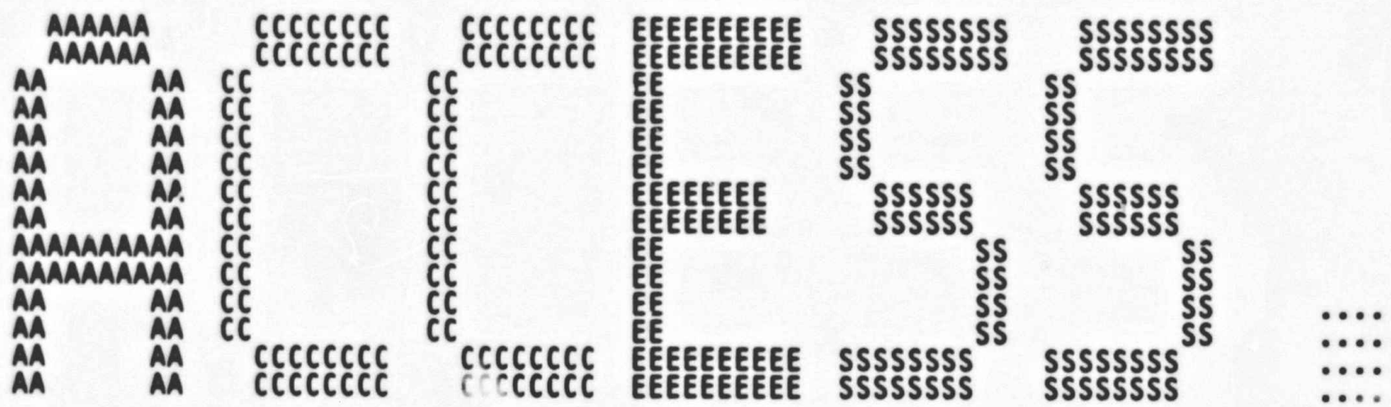
AAAAAA CCCCCCCC CCCCCCCC EEEEEEEEEE SSSSSSSS SSSSSSSS
AAAAAA CCCCCCCC CCCCCCCC EEEEEEEEEE SSSSSSSS SSSSSSSS
AA AA CC CC EE SS SS
AA AA CC CC EE SS SS
AA AA CC CC EE SS SS
AA AA CC CC EE SS SS
AA AA CC CC EEEEEEEE SSSSSS SSSSSS
AA AA CC CC EEEEEEEE SSSSSS SSSSSS
AAAAAAAAAA CC CC EE SS SS
AAAAAAAAAA CC CC EE SS SS
AA AA CC CC EE SS SS
AA AA CC CC EE SS SS
AA AA CCCCCCCC CCCCCCCC EEEEEEEEEE SSSSSSSS SSSSSSSS
AA AA CCCCCCCC CCCCCCCC EEEEEEEEEE SSSSSSSS SSSSSSSS
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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1          0001  0 MODULE ACCESS (
;     2          0002  0                 LANGUAGE (BLISS32),
;001  ! CDS0006  0003  0                 IDENT = 'V04-002'
;     4-1        0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                  *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This is the main processing routine for the ACCESS function.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !
;    41          0041  1 !       STARLET operating system, including privileged system services
;    42          0042  1 !       and internal exec routines.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  20-Dec-1976  15:43
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;001  ! CDS0006  0051  1 !       V04-002 CDS0006         Christian D. Saether    15-Nov-1984
;002  ! CDS0006  0052  1 !               Expand test for clusterness to test clu$gl_club.
;003  ! CDS0006  0053  1 !
;004  ! CDS0005  0054  1 !       V04-001 CDS0005         Christian D. Saether    14-Nov-1984
;005  ! CDS0005  0055  1 !               Make test for directory to set WRITE_TURN here
;006  ! CDS0005  0056  1 !               instead of in MAKE_ACCESS.
;007  ! CDS0005  0057  1 !
```

```
;    51          0058  1 ;        V03-023 CDS0004         Christian D. Saether    14-Aug-1984
;    52          0059  1 ;                Modify handling of extension fcbs.  Deal with stale
;    53          0060  1 ;                fcbs.
;    54          0061  1 ;
;    55          0062  1 ;        V03-022 ACG0438         Andrew C. Goldstein,    26-Jul-1984  13:41
;    56          0063  1 ;                Add interlock to special caches for write accessed
;    57          0064  1 ;                file structure files. Also move create-if handling to
;    58          0065  1 ;                the dispatcher.
;    59          0066  1 ;
;    60          0067  1 ;        V03-021 CDS0003         Christian D. Saether    20-Apr-1984
;    61          0068  1 ;                Access arbitration changes.
;    62          0069  1 ;
;    63          0070  1 ;        V03-020 ACG0412         Andrew C. Goldstein,    22-Mar-1984  18:15
;    64          0071  1 ;                Implement agent access mode support; add access mode to
;    65          0072  1 ;                check protection call
;    66          0073  1 ;
;    67          0074  1 ;        V03-019 CDS0002         Christian D. Saether     6-Mar-1984
;    68          0075  1 ;                Add re-serialization logic for coming at extension
;    69          0076  1 ;                headers directly.
;    70          0077  1 ;
;    71          0078  1 ;        V03-018 CDS001          Christian D. Saether     1-Mar-1984
;    72          0079  1 ;                Remove call to FLUSH_FID.
;    73          0080  1 ;
;    74          0081  1 ;        V03-017 CDS0009         Christian D. Saether    29-Dec-1983
;    75          0082  1 ;                Add L_NORM linkage to routine declarations.  Invoke
;    76          0083  1 ;                BASE_REGISTER and BIND_COMMON macros where needed.
;    77          0084  1 ;
;    78          0085  1 ;        V03-016 LMP0166         L. Mark Pilant,         28-Oct-1983  19:07
;    79          0086  1 ;                Correct a bug that caused execute access to grant write access
;    80          0087  1 ;                to a directory during a create-if.
;    81          0088  1 ;
;    82          0089  1 ;        V03-015 CDS0008         Christian D. Saether    23-Sep-1983
;    83          0090  1 ;                Modify interface to SERIAL_FILE routine.
;    84          0091  1 ;                Remove storing access lock ID in FIB.
;    85          0092  1 ;
;    86          0093  1 ;        V03-014 LMP0149         L. Mark Pilant,         16-Sep-1983  13:46
;    87          0094  1 ;                Fix potential buffer management problem that may occur in
;    88          0095  1 ;                READ_ATTRIB.
;    89          0096  1 ;
;    90          0097  1 ;        V03-013 ACG0354         Andrew C. Goldstein,    13-Sep-1983  16:11
;    91          0098  1 ;                Add alternate access validation mask
;    92          0099  1 ;
;    93          0100  1 ;        V03-012 CDS0007         Christian D. Saether     3-May-1983
;    94          0101  1 ;                Move ACCESS_LOCK and LOCK_MODE routines to
;    95          0102  1 ;                separate module.
;    96          0103  1 ;                Add call to SERIAL_FILE sync routine.
;    97          0104  1 ;
;    98          0105  1 ;        V03-011 CDS0006         Christian D. Saether    28-Apr-1983
;    99          0106  1 ;                Clear DELAY_TRUNC in value block if writer.
;   100          0107  1 ;
;   101          0108  1 ;        V03-010 CDS0005         Christian D. Saether    19-Apr-1983
;   102          0109  1 ;                Don't charge quota for access lock.
;   103          0110  1 ;                Bug check on unexpected errors.
;   104          0111  1 ;
;   105          0112  1 ;        V03-009 CDS0004         Christian D. Saether     6-Apr-1983
;   106          0113  1 ;                Further refinement of locking routine interfaces.
;   107          0114  1 ;                ACCESS_LOCK tests ACCLCK_ID for conversions.
```

```
;  108     0115  1 !
;  109     0116  1 !       ACCESS_LOCK tests CURRENT_UCB [UCB$L_PID] to see if shared.
;  110     0117  1 !
;  111     0118  1 !   V03-008 CDS0003         Christian D. Saether    17-Jan-1983
;  112     0119  1 !           Redo the access locking routine interface.
;  113     0120  1 !
;  114     0121  1 !   V03-007 CDS0002         Christian D. Saether     7-Jan-1983
;  115     0122  1 !           Take out access lock in exec mode.  Return lock id in fib.
;  116     0123  1 !
;  117     0124  1 !   V03-006 LMP0059         L. Mark Pilant,         21-Dec-1982  11:05
;  118     0125  1 !           Always create an FCB when a file header is accessed.  This
;  119     0126  1 !           eliminates a lot of special casing in the FCB handling.
;  120     0127  1 !
;  121     0128  1 !   V03-005 CDS0001         Christian D. Saether     6-Dec-1982
;  122     0129  1 !           Changes to support lock manager based access control.
;  123     0130  1 !
;  124     0131  1 !   V03-004 LMP48917        L. Mark Pilant,          7-Oct-1982  12:45
;  125     0132  1 !           Eliminate the explicit setting of the the access time if
;  126     0133  1 !           write access is sought for a particular file.
;  127     0134  1 !
;  128     0135  1 !   V03-003 LMP0036         L. Mark Pilant,         30-Jun-1982  10:00
;  129     0136  1 !           Add support for Access Control Lists.
;  130     0137  1 !
;  131     0138  1 !   V03-002 LMP0023         L. Mark Pilant,          8-Apr-1982  10:40
;  132     0139  1 !           If there is only one FCB, don't call REMAP_FILE but still
;  133     0140  1 !           set COMPLETE in the window.
;  134     0141  1 !
;  135     0142  1 !   V03-001 LMP0016         L. Mark Pilant,         25-Mar-1982  13:15
;  136     0143  1 !           Remove diddling of the COMPLETE bit in the window segments.
;  137     0144  1 !
;  138     0145  1 !   V02-009 ACG0258         Andrew C. Goldstein,    26-Jan-1982  16:54
;  139     0146  1 !           Fix reference to RVN 1 in expiration date processing
;  140     0147  1 !
;  141     0148  1 !   V02-008 ACG0230         Andrew C. Goldstein,    23-Dec-1981  23:17
;  142     0149  1 !           Add expiration date maintenance
;  143     0150  1 !
;  144     0151  1 !   V02-007 LMP0003         L. Mark Pilant,          8-Dec-1981  10:15
;  145     0152  1 !           Added byte limit quota check on window creation.
;  146     0153  1 !
;  147     0154  1 !   V02-006 ACG0225         Andrew C. Goldstein,    24-Nov-1981  17:18
;  148     0155  1 !           Add NOLOCK support
;  149     0156  1 !
;  150     0157  1 !   V02-005 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:24
;  151     0158  1 !           Previous revision history moved to F11B.REV
;  152     0159  1 !**
;  153     0160  1
;  154     0161  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  155     0162  1 REQUIRE 'SRC$:FCPDEF.B32';
;  156     1153  1
;  157     1154  1 FORWARD ROUTINE
;  158     1155  1         ACCESS          : L_NORM,       ! main access function processing
;  159     1156  1         SET_EXPIRE      : L_NORM;       ! enable expiration date recording
```

```
;  161     1157  1 GLOBAL ROUTINE ACCESS : L_NORM =
;  162     1158  1
;  163     1159  1 !++
;  164     1160  1 !
;  165     1161  1 ! FUNCTIONAL DESCRIPTION:
;  166     1162  1 !
;  167     1163  1 !       This is the main processing routine for the ACCESS function.
;  168     1164  1 !
;  169     1165  1 ! CALLING SEQUENCE:
;  170     1166  1 !       ACCESS ()
;  171     1167  1 !
;  172     1168  1 ! INPUT PARAMETERS:
;  173     1169  1 !       NONE
;  174     1170  1 !
;  175     1171  1 ! IMPLICIT INPUTS:
;  176     1172  1 !       CURRENT_VCB: VCB of volume
;  177     1173  1 !       IO_PACKET: address of I/O request packet
;  178     1174  1 !
;  179     1175  1 ! OUTPUT PARAMETERS:
;  180     1176  1 !       NONE
;  181     1177  1 !
;  182     1178  1 ! IMPLICIT OUTPUTS:
;  183     1179  1 !       PRIMARY_FCB: FCB of file
;  184     1180  1 !       CURRENT_WINDOW: address of file window
;  185     1181  1 !       USER_STATUS: I/O status block to return to user
;  186     1182  1 !
;  187     1183  1 ! ROUTINE VALUE:
;  188     1184  1 !       NONE
;  189     1185  1 !
;  190     1186  1 ! SIDE EFFECTS:
;  191     1187  1 !       FCB & window created
;  192     1188  1 !
;  193     1189  1 !--
;  194     1190  1
;  195     1191  2 BEGIN
;  196     1192  2
;  197     1193  2 BUILTIN
;  198     1194  2         CMPM,
;  199     1195  2         SUBM;
;  200     1196  2
;  201     1197  2 LABEL
;  202     1198  2         CHECK_EXPIRE;                   ! check file expiration date
;  203     1199  2
;  204     1200  2 LOCAL
;  205     1201  2         REALBASIS,
;  206     1202  2         STATUS,                         ! protection check status value
;  207     1203  2         FCB_CREATED,                    ! flag indicating new FCB created
;  208     1204  2         PACKET          : REF BBLOCK,   ! address of I/O packet
;  209     1205  2         ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH],
;  210     1206  2                                         ! buffer descriptors
;  211     1207  2         FIB             : REF BBLOCK,   ! file identification block
;  212     1208  2         FCB             : REF BBLOCK,   ! FCB address
;  213     1209  2         UCB             : REF BBLOCK,   ! UCB of RVN 1
;  214     1210  2         PRIMARY_VCB     : REF BBLOCK,   ! VCB of RVN 1
;  215     1211  2         HEADER          : REF BBLOCK,   ! address of file header
;  216     1212  2         NEW_HEADER      : REF BBLOCK,   ! address of extension header
;  217     1213  2         IDENT_AREA      : REF BBLOCK,   ! address of header ident area
```

```
;  218          1214  2          DAY_TIME          : VECTOR [2],   ! time of day
;  219          1215  2          FUNCTION          : BLOCK [1];    ! function code qualifiers
;  220          1216  2
;  221          1217  2  EXTERNAL
;001 !CDS0006   1218  2          CLU$GL_CLUB       : ADDRESSING_MODE (GENERAL),
;  222          1219  2          ACP$GB_WRITBACK   : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
;  223          1220  2                                                ! ACP cache writeback flags
;  224          1221  2
;  225          1222  2  BIND_COMMON;
;  226          1223  2
;  227          1224  2  EXTERNAL ROUTINE
;  228          1225  2          REBLD_PRIM_FCB    : L_NORM NOVALUE, ! rebuild primary fcb from header
;  229          1226  2          BUILD_EXT_FCBS    : L_NORM NOVALUE, ! construct extension fcbs, if nec.
;  230          1227  2          ARBITRATE_ACCESS  : L_JSB_2ARGS,    ! arbitrate file access
;  231          1228  2          CONV_ACCLOCK      : L_NORM,         ! convert access lock.
;  232          1229  2          RELEASE_SERIAL_LOCK : L_NORM NOVALUE,
;  233          1230  2          SERIAL_FILE       : L_NORM,         ! serialize file requests
;  234          1231  2          GET_FIB           : L_NORM,         ! get FIB for operation
;  235          1232  2          FIND              : L_NORM,         ! find file in directory
;  236          1233  2          CREATE            : L_NORM,         ! create file
;  237          1234  2          SWITCH_VOLUME     : L_NORM,         ! switch to correct volume
;  238          1235  2          SEARCH_FCB        : L_NORM,         ! search FCB list
;  239          1236  2          READ_HEADER       : L_NORM,         ! read file header
;  240          1237  2          CREATE_FCB        : L_NORM,         ! create an FCB
;  241          1238  2          CHECK_PROTECT     : L_NORM,         ! check file protection
;  242          1239  2          CREATE_WINDOW     : L_NORM,         ! create a window
;  243          1240  2          MAKE_ACCESS       : L_NORM ADDRESSING_MODE (GENERAL), ! complete the access
;  244          1241  2          ALLOCATION_LOCK   : L_NORM,         ! take volume allocation lock
;  245          1242  2          ALLOCATION_UNLOCK : L_NORM,         ! release volume allocation lock
;  246          1243  2          RELEASE_LOCKBASIS : L_NORM,         ! release buffers under lock
;  247          1244  2          DELETE_FID        : L_NORM,         ! flush file ID cache
;  248          1245  2          PURGE_EXTENT      : L_NORM,         ! flush extent cache
;  249          1246  2          FLUSH_QUO_CACHE   : L_NORM,         ! flush quota cache
;  250          1247  2          CACHE_LOCK        : L_NORM,         ! take out cache interlock
;  251          1248  2          CHECKSUM          : L_NORM,         ! compute file header checksum
;  252          1249  2          MARK_DIRTY        : L_NORM,         ! mark buffer for writeback
;  253          1250  2          READ_ATTRIB       : L_NORM,         ! read file attributes
;  254          1251  2          REMAP_FILE        : L_NORM,         ! remap the file completely
;  255          1252  2          MARK_COMPLETE;                      ! mark the file as complete
;  256          1253  2
;  257          1254  2  ! Enable the deaccess cleanup if an access is taking place.
;  258          1255  2  !
;  259          1256  2
;  260          1257  2  PACKET = .IO_PACKET;
;  261          1258  2  FUNCTION = .PACKET[IRP$W_FUNC];
;  262          1259  2  IF .FUNCTION[IO$V_ACCESS]
;  263          1260  2  THEN
;  264          1261  3      BEGIN
;  265          1262  3      CLEANUP_FLAGS[CLF_ZCHANNEL] = 1;
;  266          1263  3      CLEANUP_FLAGS[CLF_DELWINDOW] = 1;
;  267          1264  2      END;
;  268          1265  2
;  269          1266  2  ! Set up pointers to interesting control blocks.
;  270          1267  2  !
;  271          1268  2
;  272          1269  2                                            ! pointer to buffer descriptors
;  273          1270  2  ABD = .BBLOCK [.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
```

```
;  274     1271  2 FIB = GET_FIB (.ABD);                        ! pointer to FIB
;  275     1272  2 IF .FIB[FIB$L_ALT_ACCESS] NEQ 0
;  276     1273  2 THEN FIB[FIB$V_ALT_GRANTED] = 1;             ! assume access granted
;  277     1274  2
;  278     1275  2 ! Look up the file in the directory if called for.
;  279     1276  2 !
;  280     1277  2
;  281     1278  2 IF .CLEANUP_FLAGS[CLF_DIRECTORY]
;  282     1279  2 THEN FIND (.ABD, .FIB, 0);
;  283     1280  2
;  284     1281  2 ! If there is a file open on the channel, check the file ID returned by the
;  285     1282  2 ! FIND against the file ID that is open. If they are different, drop the FCB
;  286     1283  2 ! and window addresses on the floor.
;  287     1284  2 !
;  288     1285  2
;  289     1286  2 IF .PRIMARY_FCB NEQ 0
;  290     1287  2 THEN
;  291     1288  2     IF .PRIMARY_FCB[FCB$W_FID_NUM] NEQ .FIB[FIB$W_FID_NUM]
;  292     1289  2     OR .PRIMARY_FCB[FCB$W_FID_RVN] NEQ .FIB[FIB$W_FID_RVN]
;  293     1290  2     THEN
;  294     1291  3         BEGIN
;  295     1292  3         PRIMARY_FCB = 0;
;  296     1293  3         CURRENT_WINDOW = 0;
;  297     1294  3         END;
;  298     1295  2
;  299     1296  2 ! If this is a find only, exit now to avoid an extraneous read of the
;  300     1297  2 ! file header.
;  301     1298  2 !
;  302     1299  2
;  303     1300  2 IF NOT .FUNCTION[IO$V_ACCESS]                ! if no access
;  304     1301  2 AND .PACKET[IRP$W_BCNT] LEQ ABD$C_ATTRIB ! and no attribute list
;  305     1302  2 THEN RETURN 1;                               ! all done
;  306     1303  2
;  307     1304  2 ! Switch context to the volume of the specified RVN.
;  308     1305  2 !
;  309     1306  2
;  310     1307  2 SWITCH_VOLUME (.FIB[FIB$W_FID_RVN]);
;  311     1308  2
;  312     1309  2 ! Synchronize further processing on this file.
;  313     1310  2 !
;  314     1311  2
;  315     1312  2 PRIM_LCKINDX = SERIAL_FILE (FIB [FIB$W_FID]);
;  316     1313  2
;  317     1314  2 ! Find the FCB of the file, if one exists. then read the file
;  318     1315  2 ! header. If there is no FCB, create one.
;  319     1316  2 !
;  320     1317  2
;  321     1318  2 FCB = SEARCH_FCB (FIB[FIB$W_FID]);
;  322     1319  2 REALBASIS = 0;
;  323     1320  2 HEADER = READ_HEADER (FIB[FIB$W_FID], .FCB, REALBASIS);
;  324     1321  2
;  325     1322  2 IF .REALBASIS NEQ 0
;  326     1323  2 THEN
;  327     1324  3     BEGIN
;  328     1325  3     LOCAL
;  329     1326  3         FID : BBLOCK [6];
;  330     1327  3
```

```
;   331     1328  3      FID [FID$W_NUM] = .REALBASIS<0,16>;
;   332     1329  3      FID [FID$B_NMX] = .REALBASIS<16,8>;
;   333     1330  3      FID [FID$B_RVN] = .REALBASIS<24,8>;
;   334     1331  3
;   335     1332  3      SWITCH_VOLUME (.FID [FID$B_RVN]);
;   336     1333  3
;   337     1334  3      RELEASE_SERIAL_LOCK (.PRIM_LCKINDX);
;   338     1335  3
;   339     1336  3      PRIM_LCKINDX = SERIAL_FILE (FID);
;   340     1337  3
;   341     1338  3      IF SEARCH_FCB (FID) EQL 0
;   342     1339  3      THEN
;   343     1340  3          ERR_EXIT (SS$_NOSUCHFILE);
;   344     1341  3
;   345     1342  3      HEADER = READ_HEADER (FIB [FIB$W_FID], .FCB);
;   346     1343  3      END;
;   347     1344  1
;   348     1345  1  ! If the file is marked for delete and is not accessed by this user, and
;   349     1346  1  ! the accessor is not the system, deny its existence.
;   350     1347  1  !
;   351     1348  1
;   352     1349  1  IF .CURRENT_WINDOW EQL 0 AND .HEADER[FH2$V_MARKDEL]
;   353     1350  1  AND NOT .BBLOCK [BBLOCK [.PACKET[IRP$L_ARB], ARB$Q_PRIV], PRV$V_BYPASS]
;   354     1351  1  THEN ERR_EXIT (SS$_NOSUCHFILE);
;   355     1352  1
;   356     1353  1  FCB_CREATED = 0;
;   357     1354  1  IF .FCB EQL 0
;   358     1355  1  THEN
;   359     1356  2      BEGIN
;   360     1357  2      FCB_CREATED = 1;
;   361     1358  2      FCB = KERNEL_CALL (CREATE_FCB, .HEADER);
;   362     1359  1      END;
;   363     1360  1  PRIMARY_FCB = .FCB;                        ! record FCB for external use
;   364     1361  1
;   365     1362  1  ! If access is requested, check for conflicts and file protection.
;   366     1363  1  ! then create a window and link everything up.
;   367     1364  1  !
;   368     1365  1
;   369     1366  1  IF .FUNCTION[IO$V_ACCESS]
;   370     1367  1  THEN
;   371     1368  2      BEGIN
;   372     1369  3      IF (.HEADER[FH2$V_LOCKED])
;   373     1370  2      AND NOT .BBLOCK [BBLOCK [.PACKET[IRP$L_ARB], ARB$Q_PRIV], PRV$V_BYPASS]
;   374     1371  2      THEN ERR_EXIT (SS$_FILELOCKED);              ! file is deaccess locked
;   375     1372  2
;   376     1373  3          BEGIN
;   377     1374  3          LOCAL
;   378     1375  3              PREV_MODE;
;   379     1376  3
;   380     1377  3          PREV_MODE = .FCB [FCB$B_ACCLKMODE];
;   381     1378  3
;   382     1379  3          IF NOT ARBITRATE_ACCESS (.FIB [FIB$L_ACCTL], .FCB)
;   383     1380  3          THEN ERR_EXIT (SS$_ACCONFLICT);
;   384     1381  3
;   385     1382  3          CURRENT_WINDOW = CREATE_WINDOW (.FIB[FIB$L_ACCTL], .FIB[FIB$B_WSIZE],
;   386     1383  3                                          .HEADER, .PACKET[IRP$L_PID], .FCB);
;   387     1384  3
```

```
;   388          1385  4          IF .CURRENT_WINDOW EQL 0
;   389          1386  4          THEN
;   390          1387  5              BEGIN
;   391          1388  5
;   392          1389  5              IF .PREV_MODE<0,8> LSSU .FCB [FCB$B_ACCLKMODE]
;   393          1390  5              THEN
;   394          1391  5                  CONV_ACCLOCK (.PREV_MODE, .FCB);
;   395          1392  5
;   396          1393  5              ERR_EXIT (SS$_EXBYTLM);
;   397          1394  4              END;
;   398          1395  4
;   399          1396  4          MAKE_ACCESS (.FCB, .CURRENT_WINDOW, .ABD);
;   400          1397  4
:001  !CDS0005     1398  4          IF .HEADER [FH2$V_DIRECTORY]
:002  !CDS0005     1399  4          THEN
:003  !CDS0005     1400  5              BEGIN
:004  !CDS0005     1401  5              FCB [FCB$W_DIRSEQ] = .FCB [FCB$W_DIRSEQ] + 1;
:005  !CDS0005     1402  5              CURRENT_WINDOW [WCB$V_WRITE_TURN] = 1;
:006  !CDS0005     1403  4              END;
:007  !CDS0005     1404  4
;   401          1405  4          IF .FCB [FCB$V_DELAYTRNC]
;   402          1406  4              AND .FIB [FIB$V_WRITE]
;   403          1407  4          THEN
;   404          1408  5              BEGIN
;   405          1409  5              IF .FCB [FCB$B_ACCLKMODE] LSSU LCK$K_PWMODE
;   406          1410  5              THEN
;   407          1411  5                  IF NOT CONV_ACCLOCK (LCK$K_PWMODE, .FCB)
;   408          1412  5                  THEN
;   409          1413  5                      BUG_CHECK (XQPERR, FATAL, 'Unexpected lock manager error');
;   410          1414  5
;   411          1415  5              FCB [FCB$V_DELAYTRNC] = 0;
;   412          1416  5              FCB [FCB$L_TRUNCVBN] = 0;
;   413          1417  5
;   414          1418  5                  CONV_ACCLOCK (.FCB [FCB$B_ACCLKMODE], .FCB);
;   415          1419  4                  END;
;   416          1420  4
;   417          1421  3              END;
;   418          1422  3
;   419          1423  3      ! If file expiration is enabled and the volume is writable, check
;   420          1424  3      ! the current expiration date. If it needs to be updated, note this for
;   421          1425  3      ! processing during deaccess. Note that we use the retention parameters
;   422          1426  3      ! from RVN 1 if this is a volume set.
;   423          1427  3      !
;   424          1428  3
;   425          1429  4          CHECK_EXPIRE: BEGIN
;   426          1430  4          PRIMARY_VCB = .CURRENT_VCB;
;   427          1431  4          UCB = .CURRENT_UCB;
;   428          1432  4          IF .PRIMARY_VCB[VCB$W_RVN] NEQ 0
;   429          1433  4          THEN
;   430          1434  5              BEGIN
;   431          1435  5              UCB = .VECTOR [CURRENT_RVT[RVT$L_UCBLST], 0];
;   432          1436  5              IF .UCB EQL 0
;   433          1437  5              THEN LEAVE CHECK_EXPIRE;
;   434          1438  5              PRIMARY_VCB = .UCB[UCB$L_VCB];
;   435          1439  4              END;
;   436          1440  4
;   437          1441  4          IF NOT .BBLOCK [UCB[UCB$L_DEVCHAR], DEV$V_SWL]
```

```
;  438     1442  4      AND NOT .FIB[FIB$V_NORECORD]
;  439     1443  4      AND CMPM (2, PRIMARY_VCB[VCB$Q_RETAINMAX], UPLIT (0, 0)) NEQ 0
;  440     1444  4      AND .HEADER[FH2$B_MPOFFSET]-.HEADER[FH2$B_IDOFFSET]
;  441     1445  4          GEQU ($BYTEOFFSET (FI2$Q_EXPDATE) + FI2$S_EXPDATE) / 2
;  442     1446  4      THEN
;  443     1447  5          BEGIN
;  444     1448  5          IDENT_AREA = .HEADER + .HEADER[FH2$B_IDOFFSET]*2;
;  445     1449  5          $GETTIM (TIMADR = DAY_TIME);
;  446     1450  5          SUBQ (PRIMARY_VCB[VCB$Q_RETAINMIN], DAY_TIME);
;  447     1451  5          IF CMPM (2, IDENT_AREA[FI2$Q_EXPDATE], DAY_TIME) LSS 0
;  448     1452  5          THEN KERNEL_CALL (SET_EXPIRE);
;  449     1453  4          END;
;  450     1454  3      END;                                    ! end of block CHECK_EXPIRE
;  451     1455  3
;  452     1456  2      END;                                    ! end of access processing
;  453     1457  2
;  454     1458  2  ! If the file is multi-header, read the extension headers and create
;  455     1459  2  ! extension FCB's as necessary.
;  456     1460  2  !
;  457     1461  2
;  458     1462  2  IF .FCB_CREATED
;  459     1463  2  THEN
;  460     1464  2      BUILD_EXT_FCBS (.HEADER)
;  461     1465  2  ELSE
;  462     1466  2      IF .FCB [FCB$V_STALE]
;  463     1467  2      THEN
;  464     1468  3          BEGIN
;  465     1469  3
;  466     1470  3          REBLD_PRIM_FCB (.FCB, .HEADER);
;  467     1471  3
;  468     1472  3          BUILD_EXT_FCBS (.HEADER);
;  469     1473  3
;  470     1474  3          END;
;  471     1475  2
;  472     1476  2  !
;  473     1477  2  ! Finally make sure that access is allowed to the file.
;  474     1478  2  !
;  475     1479  2
;  476     1480  2  IF .FUNCTION[IO$V_ACCESS]
;  477     1481  2  THEN
;  478     1482  3      BEGIN
;  479     1483  3      STATUS = CHECK_PROTECT (IF .FIB[FIB$V_EXECUTE]
;  480     1484  3                              AND NOT .FIB[FIB$V_WRITE]
;  481     1485  3                              AND NOT .FIB[FIB$V_NOREAD]
;  482     1486  3                              AND .PACKET[IRP$V_MODE] LEQU 2
;  483     1487  3                              THEN EXEC_ACCESS
;  484     1488  3                              ELSE .FIB[FIB$V_WRITE] OR .FIB[FIB$V_NOREAD],
;  485     1489  3                              .HEADER,
;  486     1490  3                              .FCB,
;  487     1491  3                              MAXU (.PACKET[IRP$V_MODE], .FIB[FIB$B_AGENT_MODE]),
;  488     1492  3                              .FIB[FIB$L_ALT_ACCESS],
;  489     1493  3                              .FIB[FIB$V_ALT_REQ]);
;  490     1494  3
;  491     1495  3      IF .STATUS EQL SS$_NOTALLPRIV
;  492     1496  3      THEN FIB[FIB$V_ALT_GRANTED] = 0;
;  493     1497  3
;  494     1498  3  ! If this is a write access to the index file, the storage map, or the
```

```
;   495          1499  3 ! quota file, flush the appropriate cache. Also take out the cache lock
;   496          1500  3 ! if the volume is cluster accessible.
;   497          1501  3 !
;   498          1502  3
;   499          1503  3     IF .CURRENT_WINDOW[WCB$V_WRITE]
;   500          1504  5     AND ((.FIB[FIB$B_FID_NMX] EQL 0
;   501          1505  5           AND .FIB[FIB$W_FID_NUM] LEQU FID$C_BITMAP)
;   502          1506  4          OR .FCB EQL .CURRENT_VCB[VCB$L_QUOTAFCB])
;   503          1507  3     THEN
;   504          1508  4         BEGIN
;   505          1509  4         ALLOCATION_LOCK ();
;   506          1510  4
;   507          1511  4         IF .FCB EQL .CURRENT_VCB[VCB$L_QUOTAFCB]
;   508          1512  4         THEN FLUSH_QUO_CACHE ()
;   509          1513  4         ELSE IF .FIB[FIB$W_FID_NUM] EQL FID$C_INDEXF
;   510          1514  4         THEN DELETE_FID (0)
;   511          1515  4         ELSE PURGE_EXTENT (0, 0);
;   512          1516  4
;   513          1517  4         RELEASE_LOCKBASIS (-1);
;   514          1518  4         ALLOCATION_UNLOCK ();
;   515          1519  4
;   516          1520  4         IF .BBLOCK [CURRENT_UCB[UCB$L_DEVCHAR2], DEV$V_CLU]
;001 !CDS0006    1521  4         AND .CLU$GL_CLUB NEQ 0
;   517          1522  4         AND .FCB[FCB$L_CACHELKID] EQL 0
;   518          1523  4         THEN
;   519          1524  5             BEGIN
;   520          1525  5             STATUS = CACHE_LOCK (.FCB[FCB$L_LOCKBASIS], FCB[FCB$L_CACHELKID], 2);
;   521          1526  5             IF NOT .STATUS THEN ERR_EXIT (.STATUS);
;   522          1527  4             END;
;   523          1528  3         END;
;   524          1529  2     END;
;   525          1530  2
;   526          1531  2 ! Do read attributes if requested.
;   527          1532  2 !
;   528          1533  2
;   529          1534  2 IF .PACKET[IRP$W_BCNT] GTR ABD$C_ATTRIB
;   530          1535  2 THEN
;   531          1536  3     BEGIN
;   532          1537  3     IF .CURRENT_WINDOW EQL 0
;   533          1538  3     THEN STATUS = CHECK_PROTECT (RDATT_ACCESS, .HEADER, .FCB,
;   534          1539  3                                  MAXU (.PACKET[IRP$V_MODE], .FIB[FIB$B_AGENT_MODE]),
;   535          1540  3                                  .FIB[FIB$L_ALT_ACCESS], .FIB[FIB$V_ALT_REQ]);
;   536          1541  3     IF NOT KERNEL_CALL (READ_ATTRIB, .HEADER, .ABD) THEN ERR_EXIT ();
;   537          1542  3     HEADER = .FILE_HEADER;
;   538          1543  2     END;
;   539          1544  2
;   540          1545  2 IF .STATUS EQL     _NOTALLPRIV
;   541          1546  2 THEN FIB[FIB$V_AL _GRANTED] = 0;
;   542          1547  2
;   543          1548  2 ! If necessary map the file completely.
;   544          1549  2 !
;   545          1550  2
;   546          1551  2 IF .FUNCTION[IO$V_ACCESS]
;   547          1552  2 THEN
;   548          1553  2     IF .CURRENT_WINDOW[WCB$V_CATHEDRAL]
;   549          1554  2     THEN
;   550          1555  2         IF .PRIMARY_FCB[FCB$L_EXFCB] NEQ 0
```

```
;  551          1556  2          THEN REMAP_FILE()
;  552          1557  2          ELSE KERNEL_CALL (MARK_COMPLETE, .CURRENT_WINDOW);
;  553          1558  2
;  554          1559  2 RETURN 1;
;  555          1560  2
;  556          1561  1 END;                                    ! end of routine ACCESS

                                                                .TITLE  ACCESS
                                                                .IDENT  \V04-002\

                                                                .PSECT  $CODE$,NOWRT,2

                                00000000  00000000  00000 P.AAA:  .LONG   0, 0                                                ;

                                                                .EXTRN  CLU$GL_CLUB, ACP$GB_WRITBACK
                                                                .EXTRN  REBLD_PRIM_FCB, BUILD_EXT_FCBS
                                                                .EXTRN  ARBITRATE_ACCESS
                                                                .EXTRN  CONV_ACCLOCK, RELEASE_SERIAL_LOCK
                                                                .EXTRN  SERIAL_FILE, GET_FIB
                                                                .EXTRN  FIND, CREATE, SWITCH_VOLUME
                                                                .EXTRN  SEARCH_FCB, READ_HEADER
                                                                .EXTRN  CREATE_FCB, CHECK_PROTECT
                                                                .EXTRN  CREATE_WINDOW, MAKE_ACCESS
                                                                .EXTRN  ALLOCATION_LOCK
                                                                .EXTRN  ALLOCATION_UNLOCK
                                                                .EXTRN  RELEASE_LOCKBASIS
                                                                .EXTRN  DELETE_FID, PURGE_EXTENT
                                                                .EXTRN  FLUSH_QUO_CACHE
                                                                .EXTRN  CACHE_LOCK, CHECKSUM
                                                                .EXTRN  MARK_DIRTY, READ_ATTRIB
                                                                .EXTRN  REMAP_FILE, MARK_COMPLETE
                                                                .EXTRN  BUG$_XQPERR, SYS$GETTIM

                                                    OBFC 00000  .ENTRY  ACCESS, Save R2,R3,R4,R5,R6,R7,R8,R9,R11    ; 1157
                                              5E        14  C2 00002  SUBL2  #20, SP
                                              58  0C    AA  9E 00005  MOVAB  12(BASE), R8                           ; 1219
                                              56  90    AA  D0 00009  MOVL   -112(BASE), PACKET                     ; 1257
                                              7E  20    A6  3C 0000D  MOVZWL 32(PACKET), FUNCTION                   ; 1258
                                        06    6E        06  E1 00011  BBC    #6, FUNCTION, 1$                      ; 1259
                                           02 AA  0402  8F  A8 00015  BISW2  #1026, 2(BASE)                         ; 1263
                                              59  2C    B6  D0 0001B 1$:  MOVL   @44(PACKET), ABD                   ; 1270
                                                        59  DD 0001F  PUSHL  ABD                                    ; 1271
                                           0000G  CF        01  FB 00021  CALLS  #1, GET_FIB
                                              53            50  D0 00026  MOVL   R0, FIB
                                                    3C      A3  D5 00029  TSTL   60(FIB)                            ; 1272
                                                            04  13 0002C  BEQL   2$
                                              38  A3        02  88 0002E  BISB2  #2, 56(FIB)                       ; 1273
                                        0B    6A            06  E1 00032 2$:  BBC    #6, (BASE), 3$                ; 1278
                                                            7E  D4 00036  CLRL   -(SP)                              ; 1279
                                                            53  DD 00038  PUSHL  FIB
                                                            59  DD 0003A  PUSHL  ABD
                                           0000G  CF        03  FB 0003C  CALLS  #3, FIND
                                              50    08      AA  D0 00041 3$:  MOVL   8(BASE), R0                    ; 1286
                                                            13  13 00045  BEQL   5$
                                           04 A3    24      A0  B1 00047  CMPW   36(R0), 4(FIB)                     ; 1288
                                                            07  12 0004C  BNEQ   4$
```

```
                08   A3          28   A0   B1  0004E          CMPW    40(R0), 8(FIB)                  ; 1289
                                      05   13  00053          BEQL    5$                              ;
                                 08   AA   D4  00055  4$:     CLRL    8(BASE)                         ; 1292
                                      68   D4  00058          CLRL    (R8)                            ; 1293
          09          6E                  06   E0  0005A  5$:     BBS     #6, FUNCTION, 6$            ; 1300
                      05              32   A6   B1  0005E          CMPW    50(PACKET), #5               ; 1301
                                           03   1A  00062          BGTRU   6$                          ;
                                      0346 31   00064          BRW     46$                             ;
                      7E                  08   A3   3C  00067  6$:     MOVZWL  8(FIB), -(SP)           ; 1307
              0000G   CF                       01   FB  0006B          CALLS   #1, SWITCH_VOLUME       ;
                                           04   A3   9F  00070          PUSHAB  4(FIB)                  ; 1312
              0000G   CF                       01   FB  00073          CALLS   #1, SERIAL_FILE         ;
                18    AA                       50   D0  00078          MOVL    R0, 24(BASE)            ;
                                           04   A3   9F  0007C          PUSHAB  4(FIB)                  ; 1318
              0000G   CF                       01   FB  0007F          CALLS   #1, SEARCH_FCB          ;
                      52                       50   D0  00084          MOVL    R0, FCB                 ;
                                           04   AE   D4  00087          CLRL    REALBASIS               ; 1319
                                           04   AE   9F  0008A          PUSHAB  REALBASIS               ; 1320
                                           52   DD  0008D          PUSHL   FCB                     ;
                                           04   A3   9F  0008F          PUSHAB  4(FIB)                  ;
              0000G   CF                       03   FB  00092          CALLS   #3, READ_HEADER         ;
                      55                       50   D0  00097          MOVL    R0, HEADER              ;
                                           04   AE   D5  0009A          TSTL    REALBASIS               ; 1322
                                           45   13  0009D          BEQL    7$                          ;
                08    AE                 04   AE   B0  0009F          MOVW    REALBASIS, FID          ; 1328
                0D    AE                 06   AE   90  000A4          MOVB    REALBASIS+2, FID+5      ; 1329
                0C    AE                 07   AE   90  000A9          MOVB    REALBASIS+3, FID+4      ; 1330
                      7E                 0C   AE   9A  000AE          MOVZBL  FID+4, -(SP)            ; 1332
              0000G   CF                       01   FB  000B2          CALLS   #1, SWITCH_VOLUME       ;
                                      18   AA   DD  000B7          PUSHL   24(BASE)                ; 1334
              0000G   CF                       01   FB  000BA          CALLS   #1, RELEASE_SERIAL_LOCK ;
                                      08   AE   9F  000BF          PUSHAB  FID                     ; 1336
              0000G   CF                       01   FB  000C2          CALLS   #1, SERIAL_FILE         ;
                18    AA                       50   D0  000C7          MOVL    R0, 24(BASE)            ;
                                      08   AE   9F  000CB          PUSHAB  FID                     ; 1338
              0000G   CF                       01   FB  000CE          CALLS   #1, SEARCH_FCB          ;
                                               50   D5  000D3          TSTL    R0                      ;
                                               1B   13  000D5          BEQL    8$                      ;
                                               52   DD  000D7          PUSHL   FCB                     ; 1342
                                           04   A3   9F  000D9          PUSHAB  4(FIB)                  ;
              0000G   CF                       02   FB  000DC          CALLS   #2, READ_HEADER         ;
                      55                       50   D0  000E1          MOVL    R0, HEADER              ;
                                               68   D5  000E4  7$:     TSTL    (R8)                    ; 1349
                                               0F   12  000E6          BNEQ    9$                      ;
                                           35   A5   95  000E8          TSTB    53(HEADER)              ;
                                               0A   18  000EB          BGEQ    9$                      ;
          05        58    B6                 1D   E0  000ED          BBS     #29, @88(PACKET), 9$    ; 1350
                                      0910 8F   BF  000F2  8$:     CHMU    #2320                   ; 1351
                                               04   000F6          RET                             ;
                                               5B   D4  000F7  9$:     CLRL    FCB_CREATED             ; 1353
                                               52   D5  000F9          TSTL    FCB                     ; 1354
                                               0D   12  000FB          BNEQ    10$                     ;
                      5B                       01   D0  000FD          MOVL    #1, FCB_CREATED         ; 1357
                                               55   DD  00100          PUSHL   HEADER                  ; 1358
              0000G   CF                       01   FB  00102          CALLS   #1, CREATE_FCB          ;
                      52                       50   D0  00107          MOVL    R0, FCB                 ;
                08    AA                       52   D0  0010A  10$:     MOVL    FCB, 8(BASE)            ; 1360
```

```
                03              6E              06  E0  0010E             BBS     #6, FUNCTION, 11$                       ;  1366
                                            012D  31  00112             BRW     26$
                0A      34      A5              06  E1  00115 11$:      BBC     #6, 52(HEADER), 12$                     ;  1369
                05      58      B6              1D  E0  0011A             BBS     #29, @88(PACKET), 12$                   ;  1370
                                          08A8  8F  BF  0011F             CHMU    #2216                                   ;  1371
                                                04  00123             RET
                                        54      0B  A2  9A  00124 12$:      MOVZBL  11(FCB), PREV_MODE                      ;  1377
                                        51          52  D0  00128             MOVL    FCB, R1                         ;  1379
                                        50          63  D0  0012B             MOVL    (FIB), R0
                                                0000G   30  0012E             BSBW    ARBITRATE_ACCESS
                                        05              50  E8  00131             BLBS    R0, 13$
                                                  0800  8F  BF  00134             CHMU    #2048                           ;  1380
                                                        04  00138             RET
                                                    52  DD  00139 13$:      PUSHL   FCB                             ;  1383
                                                  0C    A6  DD  0013B             PUSHL   12(PACKET)
                                                    55  DD  0013E             PUSHL   HEADER
                                        7E      03      A3  98  00140             CVTBL   3(FIB), -(SP)                   ;  1382
                                                    63  DD  00144             PUSHL   (FIB)
                                        0000G   CF          05  FB  00146             CALLS   #5, CREATE_WINDOW
                                        68              50  D0  0014B             MOVL    R0, (R8)
                                                        68  D5  0014E             TSTL    (R8)                            ;  1385
                                                        14  12  00150             BNEQ    15$
                                          0B    A2          54  91  00152             CMPB    PREV_MODE, 11(FCB)              ;  1389
                                                        09  1E  00156             BGEQU   14$
                                                    52  DD  00158             PUSHL   FCB                             ;  1391
                                                    54  DD  0015A             PUSHL   PREV_MODE
                                        0000G   CF          02  FB  0015C             CALLS   #2, CONV_ACCLOCK
                                                  2A14  8F  BF  00161 14$:      CHMU    #10772                          ;  1393
                                                        04  00165             RET
                                                    59  DD  00166 15$:      PUSHL   ABD                             ;  1396
                                                    68  DD  00168             PUSHL   (R8)
                                                    52  DD  0016A             PUSHL   FCB
                                    00000000G   00          03  FB  0016C             CALLS   #3, MAKE_ACCESS
                                0A      35      A5          05  E1  00173             BBC     #5, 53(HEADER), 16$             ;  1398
                                              42    A2    B6  00178             INCW    66(FCB)                         ;  1401
                                                  50      68  D0  0017B             MOVL    (R8), R0                        ;  1402
                                            15    A0          10  88  0017E             BISB2   #16, 21(R0)
                                2C          23    A2          01  E1  00182 16$:      BBC     #1, 35(FCB), 18$                ;  1405
                                                  28    01    A3  E9  00187             BLBC    1(FIB), 18$                     ;  1406
                                                  04    0B    A2  91  0018B             CMPB    11(FCB), #4                     ;  1409
                                                              10  1E  0018F             BGEQU   17$
                                                              52  DD  00191             PUSHL   FCB                             ;  1411
                                                              04  DD  00193             PUSHL   #4
                                                    0000G     CF  02  FB  00195             CALLS   #2, CONV_ACCLOCK
                                                  04            50  E8  0019A             BLBS    R0, 17$
                                                          FEFF      00  0019D             BUGW                            ;  1413
                                                        0000*       9F  0019F             .WORD   <BUG$_XQPERR!4>
                                                  23    A2          02  8A  001A1 17$:      BICB2   #2, 35(FCB)             ;  1415
                                                              50    A2  D4  001A5             CLRL    80(FCB)                 ;  1416
                                                              52  DD  001A8             PUSHL   FCB                             ;  1418
                                                  7E      0B  A2  9A  001AA             MOVZBL  11(FCB), -(SP)
                                                0000G     CF  02  FB  001AE             CALLS   #2, CONV_ACCLOCK
                                                    54      98  AA  D0  001B3 18$:      MOVL    -104(BASE), PRIMARY_VCB         ;  1430
                                                    50      94  AA  D0  001B7             MOVL    -108(BASE), UCB                 ;  1431
                                                            0E  A4  B5  001BB             TSTW    14(PRIMARY_VCB)                 ;  1432
                                                            0E  13  001BE             BEQL    19$
                                                    51      9C  AA  D0  001C0             MOVL    -100(BASE), R1                  ;  1435
```

```
                        50      44   A1  D0 001C4          MOVL    68(R1), UCB                                     ;
                                     78  13 001C8          BEQL    26$                                             ;
                        54      34   A0  D0 001CA          MOVL    52(UCB), PRIMARY_VCB                            ; 1436
6F                      A0           01  E0 001CE 19$:     BBS     #1, 59(UCB), 26$                                ; 1438
6B                      63           15  E0 001D3          BBS     #21, (FIB), 26$                                 ; 1441
                        50           01  CE 001D7          MNEGL   #1, R0                                          ; 1442
                  FE1C  CF      78   A4  D1 001DA          CMPL    120(PRIMARY_VCB), P.AAA+4                       ; 1443
                                     10  19 001E0          BLSS    22$                                             ;
                                     0A  14 001E2          BGTR    20$                                             ;
                  FE0E  CF      74   A4  D1 001E4          CMPL    116(PRIMARY_VCB), P.AAA                         ;
                                     04  13 001EA          BEQL    21$                                             ;
                                     04  1F 001EC          BLSSU   22$                                             ;
                                     50  D6 001EE 20$:      INCL    R0                                              ;
                                     50  D6 001F0 21$:      INCL    R0                                              ;
                                     50  D5 001F2 22$:      TSTL    R0                                              ;
                                     4C  13 001F4          BEQL    26$                                             ;
                        50      01   A5  9A 001F6          MOVZBL  1(HEADER), R0                                   ; 1444
                        51           65  9A 001FA          MOVZBL  (HEADER), R1                                    ;
                        50           51  C2 001FD          SUBL2   R1, R0                                          ;
                        17           50  D1 00200          CMPL    R0, #23                                         ; 1445
                                     3D  1F 00203          BLSSU   26$                                             ;
                        50           65  9A 00205          MOVZBL  (HEADER), R0                                    ; 1448
                        57         6540  3E 00208          MOVAW   (HEADER)[R0], IDENT_AREA                        ;
                                     10  AE 9F 0020C          PUSHAB  DAY_TIME                                        ; 1449
            00000000G   00           01  FB 0020F          CALLS   #1, SYS$GETTIM                                  ;
                  10    AE      6C   A4  C2 00216          SUBL2   108(PRIMARY_VCB), DAY_TIME                      ; 1450
                  14    AE      70   A4  D9 0021B          SBWC    112(PRIMARY_VCB), DAY_TIME                      ;
                        50           01  CE 00220          MNEGL   #1, R0                                          ; 1451
                  14    AE      2A   A7  D1 00223          CMPL    42(IDENT_AREA), DAY_TIME                        ;
                                     0F  19 00228          BLSS    25$                                             ;
                                     09  14 0022A          BGTR    23$                                             ;
                  10    AE      26   A7  D1 0022C          CMPL    38(IDENT_AREA), DAY_TIME                        ;
                                     04  13 00231          BEQL    24$                                             ;
                                     04  1F 00233          BLSSU   25$                                             ;
                                     50  D6 00235 23$:      INCL    R0                                              ;
                                     50  D6 00237 24$:      INCL    R0                                              ;
                                     50  D5 00239 25$:      TSTL    R0                                              ;
                                     05  1F 0023B          BGEQ    26$                                             ;
                  0000V CF           00  FB 0023D          CALLS   #0, SET_EXPIRE                                  ; 1452
                        0B           5B  E8 00242 26$:      BLBS    FCB_CREATED, 27$                                ; 1462
                        0E      23   A2  E9 00245          BLBC    35(FCB), 28$                                    ; 1466
                                     24  BB 00249          PUSHR   #^M<R2,R5>                                      ; 1470
                  0000G CF           02  FB 0024B          CALLS   #2, REBLD_PRIM_FCB                              ;
                                     55  DD 00250 27$:      PUSHL   HEADER                                          ; 1472
                  0000G CF           01  FB 00252          CALLS   #1, BUILD_EXT_FCBS                              ;
                     03           6E   06  E0 00257 28$:      BBS     #6, FUNCTION, 29$                               ; 1480
                                  00DB  31 0025B          BRW     39$                                             ;
7E          38  A3                01   00  EF 0025E 29$:      EXTZV   #0, #1, 56(FIB), -(SP)                          ; 1493
                                     3C  A3  DD 00264          PUSHL   60(FIB)                                         ; 1492
7E          0B  A6                02   00  EF 00267          EXTZV   #0, #2, 11(PACKET), -(SP)                       ; 1491
                        6E      2E   A3  91 0026D          CMPB    46(FIB), (SP)                                   ;
                                     04  1B 00271          BLEQU   30$                                             ;
                        6E      2E   A3  9A 00273          MOVZBL  46(FIB), (SP)                                   ;
                                     52  DD 00277 30$:      PUSHL   FCB                                             ; 1490
                                     55  DD 00279          PUSHL   HEADER                                          ; 1489
                        14      02   A3  E9 0027B          BLBC    2(FIB), 31$                                     ; 1483
                        10      01   A3  E8 0027F          BLBS    1(FIB), 31$                                     ; 1484
```

```
                  0C               63  0A  E0 00283          BBS     #10, (FIB), 31$                       ; 1485
02        0B      A6               02  00  ED 00287          CMPZV   #0, #2, 11(PACKET), #2                ; 1486
                                       04  1A 0028D          BGTRU   31$
                                       06  DD 0028F          PUSHL   #6                                    ; 1483
                                       10  11 00291          BRB     32$
50        01      A3               01  00  EF 00293  31$:    EXTZV   #0, #1, 1(FIB), R0                    ; 1488
51                63               01  0A  EF 00299          EXTZV   #10, #1, (FIB), R1
                                   50  51  C8 0029E          BISL2   R1, R0
                                       50  DD 002A1          PUSHL   R0
                        0000G  CF      06  FB 002A3  32$:    CALLS   #6, CHECK_PROTECT                     ; 1483
                               57      50  D0 002A8          MOVL    R0, STATUS
                     00000681  8F      57  D1 002AB          CMPL    STATUS, #1665                         ; 1495
                                       04  12 002B2          BNEQ    33$
                           38  A3      02  8A 002B4          BICB2   #2, 56(FIB)                           ; 1496
                               50      68  D0 002B8  33$:    MOVL    (R8), R0                              ; 1503
                  79       0B  A0      01  E1 002BB          BBC     #1, 11(R0), 39$
                                   09  A3  95 002C0          TSTB    9(FIB)                                ; 1504
                                       06  12 002C3          BNEQ    34$
                               02  04  A3  B1 002C5          CMPW    4(FIB), #2                            ; 1505
                                       0A  1B 002C9          BLEQU   35$
                               50  98  AA  D0 002CB  34$:    MOVL    -104(BASE), R0                        ; 1506
                           54  A0      52  D1 002CF          CMPL    FCB, 84(R0)
                                       64  12 002D3          BNEQ    39$
                        0000G  CF      00  FB 002D5  35$:    CALLS   #0, ALLOCATION_LOCK                   ; 1509
                               50  98  AA  D0 002DA          MOVL    -104(BASE), R0                        ; 1511
                           54  A0      52  D1 002DE          CMPL    FCB, 84(R0)
                                       07  12 002E2          BNEQ    36$
                        0000G  CF      00  FB 002E4          CALLS   #0, FLUSH_QUO_CACHE                   ; 1512
                                       16  11 002E9          BRB     38$
                               01  04  A3  B1 002EB  36$:    CMPW    4(FIB), #1                            ; 1513
                                       09  12 002EF          BNEQ    37$
                                       7E  D4 002F1          CLRL    -(SP)                                 ; 1514
                        0000G  CF      01  FB 002F3          CALLS   #1, DELETE_FID
                                       07  11 002F8          BRB     38$
                                       7E  7C 002FA  37$:    CLRQ    -(SP)                                 ; 1515
                        0000G  CF      02  FB 002FC          CALLS   #2, PURGE_EXTENT
                               7E      01  CE 00301  38$:    MNEGL   #1, -(SP)                             ; 1517
                        0000G  CF      01  FB 00304          CALLS   #1, RELEASE_LOCKBASIS
                        0000G  CF      00  FB 00309          CALLS   #0, ALLOCATION_UNLOCK                 ; 1518
                               50  94  AA  D0 0030E          MOVL    -108(BASE), R0                        ; 1520
                               23  3C  A0  E9 00312          BLBC    60(R0), 39$
                            00000000G  00  D5 00316          TSTL    CLU$GL_CLUB                           ; 1521
                                       1B  13 0031C          BEQL    39$
                                   54  A2  D5 0031E          TSTL    84(FCB)                               ; 1522
                                       16  12 00321          BNEQ    39$
                                       02  DD 00323          PUSHL   #2                                    ; 1525
                                   54  A2  9F 00325          PUSHAB  84(FCB)
                                   4C  A2  DD 00328          PUSHL   76(FCB)
                        0000G  CF      03  FB 0032B          CALLS   #3, CACHE_LOCK
                               57      50  D0 00330          MOVL    R0, STATUS
                               03      57  E8 00333          BLBS    STATUS, 39$                           ; 1526
                                       57  BF 00336          CHMU    STATUS
                                           04 00338          RET
                               05  32  A6  B1 00339  39$:    CMPW    50(PACKET), #5                        ; 1534
                                       3E  1B 0033D          BLEQU   43$
                                       68  D5 0033F          TSTL    (R8)                                  ; 1537
                                       27  12 00341          BNEQ    41$
```

```
                                                00 EF 00343          EXTZV   #0, #1, 56(FIB), -(SP)                ; 1540
                                             3C A3 DD 00349          PUSHL   60(FIB)
                                                00 EF 0034C          EXTZV   #0, #2, 11(PACKET), -(SP)             ; 1539
                                             2E A3 91 00352          CMPB    46(FIB), (SP)
                                                04 1B 00356          BLEQU   40$
                                             2E A3 9A 00358          MOVZBL  46(FIB), (SP)
                                                52 DD 0035C 40$:     PUSHL   FCB                                   ; 1538
                                                55 DD 0035E          PUSHL   HEADER
                                                04 DD 00360          PUSHL   #4
                                          0000G CF 06 FB 00362       CALLS   #6, CHECK_PROTECT
                                                57 50 DD 00367       MOVL    R0, STATUS
                                          0220 8F BB 0036A 41$:      PUSHR   #^M<R5,R9>                            ; 1541
                                          0000G CF 02 FB 0036E       CALLS   #2, READ_ATTRIB
                                                03 50 E8 00373       BLBS    R0, 42$
                                                   00 BF 00376       CHMU    #0
                                                   04 00378          RET
                                                55 04 AA D0 00379 42$: MOVL    4(BASE), HEADER                        ; 1542
                                    00000681 8F    57 D1 0037D 43$:    CMPL    STATUS, #1665                      ; 1545
                                                   04 12 00384         BNEQ    44$
                                          38 A3    02 8A 00386         BICB2   #2, 56(FIB)                        ; 1546
                                       1F 6E       06 E1 0038A 44$:    BBC     #6, FUNCTION, 46$                  ; 1551
                                          51       68 D0 0038E         MOVL    (R8), R1                           ; 1553
                                    17 0B A1       06 E1 00391         BBC     #6, 11(R1), 46$
                                          50    08 AA D0 00396         MOVL    8(BASE), R0                        ; 1555
                                             0C A0 D5 0039A            TSTL    12(R0)
                                                07 13 0039D          BEQL    45$
                                          0000G CF 00 FB 0039F       CALLS   #0, REMAP_FILE                        ; 1556
                                                07 11 003A4          BRB     46$
                                                51 DD 003A6 45$:     PUSHL   R1                                    ; 1557
                                          0000G CF 01 FB 003A8       CALLS   #1, MARK_COMPLETE
                                                50 01 D0 003AD 46$:  MOVL    #1, R0                             ; 1559
                                                   04 003B0          RET                                           ; 1561

; Routine Size:  945 bytes,    Routine Base:  $CODE$ + 0008
```

```
;  558          1562  1 GLOBAL ROUTINE SET_EXPIRE : L_NORM =
;  559          1563  1
;  560          1564  1 !++
;  561          1565  1 !
;  562          1566  1 ! FUNCTIONAL DESCRIPTION:
;  563          1567  1 !
;  564          1568  1 !       This routine sets the bit in a window marking the file for
;  565          1569  1 !       expiration date recording when it is closed.
;  566          1570  1 !
;  567          1571  1 ! CALLING SEQUENCE:
;  568          1572  1 !       SET_EXPIRE ()
;  569          1573  1 !
;  570          1574  1 ! INPUT PARAMETERS:
;  571          1575  1 !       NONE
;  572          1576  1 !
;  573          1577  1 ! IMPLICIT INPUTS:
;  574          1578  1 !       CURRENT_WINDOW: address of file window
;  575          1579  1 !
;  576          1580  1 ! OUTPUT PARAMETERS:
;  577          1581  1 !       NONE
;  578          1582  1 !
;  579          1583  1 ! IMPLICIT OUTPUTS:
;  580          1584  1 !       NONE
;  581          1585  1 !
;  582          1586  1 ! ROUTINE VALUE:
;  583          1587  1 !       1
;  584          1588  1 !
;  585          1589  1 ! SIDE EFFECTS:
;  586          1590  1 !       expire bit set
;  587          1591  1 !
;  588          1592  1 !--
;  589          1593  1
;  590          1594  2 BEGIN
;  591          1595  2
;  592          1596  2 BIND_COMMON;
;  593          1597  2
;  594          1598  2 CURRENT_WINDOW[WCB$V_EXPIRE] = 1;
;  595          1599  2
;  596          1600  2 1
;  597          1601  1 END;                                    ! End of routine SET_EXPIRE


                                        0000 00000      .ENTRY  SET_EXPIRE, Save nothing                ; 1562
                              50    0C  AA   D0 00002   MOVL    12(BASE), R0                            ; 1598
                        0B    A0    80  8F   88 00006   BISB2   #128, 11(R0)                            ;
                              50        01   D0 0000B   MOVL    #1, R0                                  ; 1601
                                         04 0000E       RET                                             ;

; Routine Size:  15 bytes,    Routine Base:  $CODE$ + 03B9

;  598          1602  1
;  599          1603  1 END
;  600          1604  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                        968  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                 Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA18:[SYSLIB]LIB.L32;1            18619        89         0        1000        00:02.0


;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:ACCESS/OBJ=OBJ$:ACCESS MSRC$:ACCESS/UPDATE=(BUG$:ACCESS)

; Size:         960 code + 8 data bytes
; Run Time:       00:42.4
; Elapsed Time:   01:07.6
; Lines/CPU Min:   2271
; Lexemes/CPU-Min: 40570
; Memory Used:  434 pages
; Compilation Complete
```

0442 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

F11BXQP
MAP

ACCESS
LIS

TA78DVDEP
LIS

CLENUP
LIS

CHARGEQ
LIS

F11X