F11X

```
_$25
Symb
----
IOC$
IO_C
IO_C
IO_D
IO_F
IO_S
KILL

KILL
KILL
LB_E
LB_D
LB_F
LB_H
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK
```

FILEID**ACCESS

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

```
;     1          0001  0 MODULE ACCESS (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  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 !
;    51          0051  1 !       V03-023 CDS0004         Christian D. Saether    14-Aug-1984
;    52          0052  1 !               Modify handling of extension fcbs.  Deal with stale
;    53          0053  1 !               fcbs.
;    54          0054  1 !
;    55          0055  1 !       V03-022 ACG0438         Andrew C. Goldstein,    26-Jul-1984  13:41
;    56          0056  1 !               Add interlock to special caches for write accessed
;    57          0057  1 !               file structure files. Also move create-if handling to
```

```
;    58          0058  1 !                          the dispatcher.
;    59          0059  1 !
;    60          0060  1 !                  V03-021 CDS0003         Christian D. Saether    20-Apr-1984
;    61          0061  1 !                          Access arbitration changes.
;    62          0062  1 !
;    63          0063  1 !                  V03-020 ACG0412         Andrew C. Goldstein,    22-Mar-1984  18:15
;    64          0064  1 !                          Implement agent access mode support; add access mode to
;    65          0065  1 !                          check protection call
;    66          0066  1 !
;    67          0067  1 !                  V03-019 CDS0002         Christian D. Saether     6-Mar-1984
;    68          0068  1 !                          Add re-serialization logic for coming at extension
;    69          0069  1 !                          headers directly.
;    70          0070  1 !
;    71          0071  1 !                  V03-018 CDS001          Christian D. Saether     1-Mar-1984
;    72          0072  1 !                          Remove call to FLUSH_FID.
;    73          0073  1 !
;    74          0074  1 !                  V03-017 CDS0009         Christian D. Saether    29-Dec-1983
;    75          0075  1 !                          Add L_NORM linkage to routine declarations.  Invoke
;    76          0076  1 !                          BASE_REGISTER and BIND_COMMON macros where needed.
;    77          0077  1 !
;    78          0078  1 !                  V03-016 LMP0166         L. Mark Pilant,         28-Oct-1983  19:07
;    79          0079  1 !                          Correct a bug that caused execute access to grant write access
;    80          0080  1 !                          to a directory during a create-if.
;    81          0081  1 !
;    82          0082  1 !                  V03-015 CDS0008         Christian D. Saether    23-Sep-1983
;    83          0083  1 !                          Modify interface to SERIAL_FILE routine.
;    84          0084  1 !                          Remove storing access lock ID in FIB.
;    85          0085  1 !
;    86          0086  1 !                  V03-014 LMP0149         L. Mark Pilant,         16-Sep-1983  13:46
;    87          0087  1 !                          Fix potential buffer management problem that may occur in
;    88          0088  1 !                          READ_ATTRIB.
;    89          0089  1 !
;    90          0090  1 !                  V03-013 ACG0354         Andrew C. Goldstein,    13-Sep-1983  16:11
;    91          0091  1 !                          Add alternate access validation mask
;    92          0092  1 !
;    93          0093  1 !                  V03-012 CDS0007         Christian D. Saether     3-May-1983
;    94          0094  1 !                          Move ACCESS_LOCK and LOCK_MODE routines to
;    95          0095  1 !                          separate module.
;    96          0096  1 !                          Add call to SERIAL_FILE sync routine.
;    97          0097  1 !
;    98          0098  1 !                  V03-011 CDS0006         Christian D. Saether    28-Apr-1983
;    99          0099  1 !                          Clear DELAY_TRUNC in value block if writer.
;   100          0100  1 !
;   101          0101  1 !                  V03-010 CDS0005         Christian D. Saether    19-Apr-1983
;   102          0102  1 !                          Don't charge quota for access lock.
;   103          0103  1 !                          Bug check on unexpected errors.
;   104          0104  1 !
;   105          0105  1 !                  V03-009 CDS0004         Christian D. Saether     6-Apr-1983
;   106          0106  1 !                          Further refinement of locking routine interfaces.
;   107          0107  1 !                          ACCESS_LOCK tests ACCLCK_ID for conversions.
;   108          0108  1 !                          ACCESS_LOCK tests CURRENT_UCB [UCB$L_PID] to see if shared.
;   109          0109  1 !
;   110          0110  1 !                  V03-008 CDS0003         Christian D. Saether    17-Jan-1983
;   111          0111  1 !                          Redo the access locking routine interface.
;   112          0112  1 !
;   113          0113  1 !                  V03-007 CDS0002         Christian D. Saether     7-Jan-1983
;   114          0114  1 !                          Take out access lock in exec mode.  Return lock id in fib.
```

```
;  115     0115  1 !
;  116     0116  1 !       V03-006 LMP0059         L. Mark Pilant,         21-Dec-1982  11:05
;  117     0117  1 !               Always create an FCB when a file header is accessed.  This
;  118     0118  1 !               eliminates a lot of special casing in the FCB handling.
;  119     0119  1 !
;  120     0120  1 !       V03-005 CDS0001         Christian D. Saether     6-Dec-1982
;  121     0121  1 !               Changes to support lock manager based access control.
;  122     0122  1 !
;  123     0123  1 !       V03-004 LMP48917        L. Mark Pilant,         7-Oct-1982  12:45
;  124     0124  1 !               Eliminate the explicit setting of the the access time if
;  125     0125  1 !               write access is sought for a particular file.
;  126     0126  1 !
;  127     0127  1 !       V03-003 LMP0036         L. Mark Pilant,         30-Jun-1982  10:00
;  128     0128  1 !               Add support for Access Control Lists.
;  129     0129  1 !
;  130     0130  1 !       V03-002 LMP0023         L. Mark Pilant,         8-Apr-1982  10:40
;  131     0131  1 !               If there is only one FCB, don't call REMAP_FILE but still
;  132     0132  1 !               set COMPLETE in the window.
;  133     0133  1 !
;  134     0134  1 !       V03-001 LMP0016         L. Mark Pilant,         25-Mar-1982  13:15
;  135     0135  1 !               Remove diddling of the COMPLETE bit in the window segments.
;  136     0136  1 !
;  137     0137  1 !       V02-009 ACG0258         Andrew C. Goldstein,    26-Jan-1982  16:54
;  138     0138  1 !               Fix reference to RVN 1 in expiration date processing
;  139     0139  1 !
;  140     0140  1 !       V02-008 ACG0230         Andrew C. Goldstein,    23-Dec-1981  23:17
;  141     0141  1 !               Add expiration date maintenance
;  142     0142  1 !
;  143     0143  1 !       V02-007 LMP0003         L. Mark Pilant,         8-Dec-1981  10:15
;  144     0144  1 !               Added byte limit quota check on window creation.
;  145     0145  1 !
;  146     0146  1 !       V02-006 ACG0225         Andrew C. Goldstein,    24-Nov-1981  17:18
;  147     0147  1 !               Add NOLOCK support
;  148     0148  1 !
;  149     0149  1 !       V02-005 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:24
;  150     0150  1 !               Previous revision history moved to F11B.REV
;  151     0151  1 !**
;  152     0152  1
;  153     0153  1
;  154     0154  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  155     0155  1 REQUIRE 'SRC$:FCPDEF.B32';
;  156     1146  1
;  157     1147  1 FORWARD ROUTINE
;  158     1148  1         ACCESS          : L_NORM,       ! main access function processing
;  159     1149  1         SET_EXPIRE      : L_NORM;       ! enable expiration date recording
```

```
;  161          1150  1 GLOBAL ROUTINE ACCESS : L_NORM =
;  162          1151  1
;  163          1152  1 !++
;  164          1153  1 !
;  165          1154  1 ! FUNCTIONAL DESCRIPTION:
;  166          1155  1 !
;  167          1156  1 !       This is the main processing routine for the ACCESS function.
;  168          1157  1 !
;  169          1158  1 ! CALLING SEQUENCE:
;  170          1159  1 !       ACCESS ()
;  171          1160  1 !
;  172          1161  1 ! INPUT PARAMETERS:
;  173          1162  1 !       NONE
;  174          1163  1 !
;  175          1164  1 ! IMPLICIT INPUTS:
;  176          1165  1 !       CURRENT_VCB: VCB of volume
;  177          1166  1 !       IO_PACKET: address of I/O request packet
;  178          1167  1 !
;  179          1168  1 ! OUTPUT PARAMETERS:
;  180          1169  1 !       NONE
;  181          1170  1 !
;  182          1171  1 ! IMPLICIT OUTPUTS:
;  183          1172  1 !       PRIMARY_FCB: FCB of file
;  184          1173  1 !       CURRENT_WINDOW: address of file window
;  185          1174  1 !       USER_STATUS: I/O status block to return to user
;  186          1175  1 !
;  187          1176  1 ! ROUTINE VALUE:
;  188          1177  1 !       NONE
;  189          1178  1 !
;  190          1179  1 ! SIDE EFFECTS:
;  191          1180  1 !       FCB & window created
;  192          1181  1 !
;  193          1182  1 !--
;  194          1183  1
;  195          1184  2 BEGIN
;  196          1185  2
;  197          1186  2 BUILTIN
;  198          1187  2         CMPM,
;  199          1188  2         SUBM;
;  200          1189  2
;  201          1190  2 LABEL
;  202          1191  2         CHECK_EXPIRE;                   ! check file expiration date
;  203          1192  2
;  204          1193  2 LOCAL
;  205          1194  2         REALBASIS,
;  206          1195  2         STATUS,                         ! protection check status value
;  207          1196  2         FCB_CREATED,                    ! flag indicating new FCB created
;  208          1197  2         PACKET          : REF BBLOCK,   ! address of I/O packet
;  209          1198  2         ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH],
;  210          1199  2                                         ! buffer descriptors
;  211          1200  2         FIB             : REF BBLOCK,   ! file identification block
;  212          1201  2         FCB             : REF BBLOCK,   ! FCB address
;  213          1202  2         UCB             : REF BBLOCK,   ! UCB of RVN 1
;  214          1203  2         PRIMARY_VCB     : REF BBLOCK,   ! VCB of RVN 1
;  215          1204  2         HEADER          : REF BBLOCK,   ! address of file header
;  216          1205  2         NEW_HEADER      : REF BBLOCK,   ! address of extension header
;  217          1206  2         IDENT_AREA      : REF BBLOCK,   ! address of header ident area
```

```
;  218          1207  2          DAY_TIME        : VECTOR [2],    ! time of day
;  219          1208  2          FUNCTION        : BLOCK [1];     ! function code qualifiers
;  220          1209  2
;  221          1210  2 EXTERNAL
;  222          1211  2          ACP$GB_WRITBACK : BITVECTOR ADDRESSING_MODE (ABSOLUTE);
;  223          1212  2                                           ! ACP cache writeback flags
;  224          1213  2
;  225          1214  2 BIND_COMMON;
;  226          1215  2
;  227          1216  2 EXTERNAL ROUTINE
;  228          1217  2          REBLD_PRIM_FCB  : L_NORM NOVALUE, ! rebuild primary fcb from header
;  229          1218  2          BUILD_EXT_FCBS  : L_NORM NOVALUE, ! construct extension fcbs, if nec.
;  230          1219  2          ARBITRATE_ACCESS : [_JSB_2ARGS, ! arbitrate file access
;  231          1220  2          CONV_ACCLOCK    : L_NORM,        ! convert access lock.
;  232          1221  2          RELEASE_SERIAL_LOCK : L_NORM NOVALUE,
;  233          1222  2          SERIAL_FILE     : L_NORM,        ! serialize file requests
;  234          1223  2          GET_FIB         : L_NORM,        ! get FIB for operation
;  235          1224  2          FIND            : L_NORM,        ! find file in directory
;  236          1225  2          CREATE          : L_NORM,        ! create file
;  237          1226  2          SWITCH_VOLUME   : L_NORM,        ! switch to correct volume
;  238          1227  2          SEARCH_FCB      : L_NORM,        ! search FCB list
;  239          1228  2          READ_HEADER     : L_NORM,        ! read file header
;  240          1229  2          CREATE_FCB      : L_NORM,        ! create an FCB
;  241          1230  2          CHECK_PROTECT   : L_NORM,        ! check file protection
;  242          1231  2          CREATE_WINDOW   : L_NORM,        ! create a window
;  243          1232  2          MAKE_ACCESS     : L_NORM ADDRESSING_MODE (GENERAL), ! complete the access
;  244          1233  2          ALLOCATION_LOCK : L_NORM,        ! take volume allocation lock
;  245          1234  2          ALLOCATION_UNLOCK : L_NORM,      ! release volume allocation lock
;  246          1235  2          RELEASE_LOCKBASIS : L_NORM,      ! release buffers under lock
;  247          1236  2          DELETE_FID      : L_NORM,        ! flush file ID cache
;  248          1237  2          PURGE_EXTENT    : L_NORM,        ! flush extent cache
;  249          1238  2          FLUSH_QUO_CACHE : L_NORM,        ! flush quota cache
;  250          1239  2          CACHE_LOCK      : L_NORM,        ! take out cache interlock
;  251          1240  2          CHECKSUM        : L_NORM,        ! compute file header checksum
;  252          1241  2          MARK_DIRTY      : L_NORM,        ! mark buffer for writeback
;  253          1242  2          READ_ATTRIB     : L_NORM,        ! read file attributes
;  254          1243  2          REMAP_FILE      : L_NORM,        ! remap the file completely
;  255          1244  2          MARK_COMPLETE;                   ! mark the file as complete
;  256          1245  2
;  257          1246  2 ! Enable the deaccess cleanup if an access is taking place.
;  258          1247  2 !
;  259          1248  2
;  260          1249  2 PACKET = .IO_PACKET;
;  261          1250  2 FUNCTION = .PACKET[IRP$W_FUNC];
;  262          1251  2 IF .FUNCTION[IO$V_ACCESS]
;  263          1252  2 THEN
;  264          1253  3     BEGIN
;  265          1254  3     CLEANUP_FLAGS[CLF_ZCHANNEL] = 1;
;  266          1255  3     CLEANUP_FLAGS[CLF_DELWINDOW] = 1;
;  267          1256  2     END;
;  268          1257  2
;  269          1258  2 ! Set up pointers to interesting control blocks.
;  270          1259  2 !
;  271          1260  2
;  272          1261  2                                           ! pointer to buffer descriptors
;  273          1262  2 ABD = .BBLOCK [.PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;  274          1263  2 FIB = GET_FIB (.ABD);                     ! pointer to FIB
```

```
;  275          1264  2 IF .FIB[FIB$L_ALT_ACCESS] NEQ 0
;  276          1265  2 THEN FIB[FIB$V_ALT_GRANTED] = 1;        ! assume access granted
;  277          1266  2
;  278          1267  2 ! Look up the file in the directory if called for.
;  279          1268  2 !
;  280          1269  2
;  281          1270  2 IF .CLEANUP_FLAGS[CLF_DIRECTORY]
;  282          1271  2 THEN FIND (.ABD, .FIB, 0);
;  283          1272  2
;  284          1273  2 ! If there is a file open on the channel, check the file ID returned by the
;  285          1274  2 ! FIND against the file ID that is open. If they are different, drop the FCB
;  286          1275  2 ! and window addresses on the floor.
;  287          1276  2 !
;  288          1277  2
;  289          1278  2 IF .PRIMARY_FCB NEQ 0
;  290          1279  2 THEN
;  291          1280  2     IF .PRIMARY_FCB[FCB$W_FID_NUM] NEQ .FIB[FIB$W_FID_NUM]
;  292          1281  2     OR .PRIMARY_FCB[FCB$W_FID_RVN] NEQ .FIB[FIB$W_FID_RVN]
;  293          1282  2     THEN
;  294          1283  3         BEGIN
;  295          1284  3         PRIMARY_FCB = 0;
;  296          1285  3         CURRENT_WINDOW = 0;
;  297          1286  2         END;
;  298          1287  2
;  299          1288  2 ! If this is a find only, exit now to avoid an extraneous read of the
;  300          1289  2 ! file header.
;  301          1290  2 !
;  302          1291  2
;  303          1292  2 IF NOT .FUNCTION[IO$V_ACCESS]           ! if no access
;  304          1293  2 AND .PACKET[IRP$W_BCNT] LEQ ABD$C_ATTRIB ! and no attribute list
;  305          1294  2 THEN RETURN 1;                          ! all done
;  306          1295  2
;  307          1296  2 ! Switch context to the volume of the specified RVN.
;  308          1297  2 !
;  309          1298  2
;  310          1299  2 SWITCH_VOLUME (.FIB[FIB$W_FID_RVN]);
;  311          1300  2
;  312          1301  2 ! Synchronize further processing on this file.
;  313          1302  2 !
;  314          1303  2
;  315          1304  2 PRIM_LCKINDX = SERIAL_FILE (FIB [FIB$W_FID]);
;  316          1305  2
;  317          1306  2 ! Find the FCB of the file, if one exists. then read the file
;  318          1307  2 ! header. If there is no FCB, create one.
;  319          1308  2 !
;  320          1309  2
;  321          1310  2 FCB = SEARCH_FCB (FIB[FIB$W_FID]);
;  322          1311  2 REALBASIS = 0;
;  323          1312  2 HEADER = READ_HEADER (FIB[FIB$W_FID], .FCB, REALBASIS);
;  324          1313  2
;  325          1314  2 IF .REALBASIS NEQ 0
;  326          1315  2 THEN
;  327          1316  3     BEGIN
;  328          1317  3     LOCAL
;  329          1318  3         FID : BBLOCK [6];
;  330          1319  3
;  331          1320  3     FID [FID$W_NUM] = .REALBASIS<0,16>;
```

```
;  332          1321  3      FID [FID$B_NMX] = .REALBASIS<16,8>;
;  333          1322  3      FID [FID$B_RVN] = .REALBASIS<24,8>;
;  334          1323  3
;  335          1324  3      SWITCH_VOLUME (.FID [FID$B_RVN]);
;  336          1325  3
;  337          1326  3      RELEASE_SERIAL_LOCK (.PRIM_LCKINDX);
;  338          1327  3
;  339          1328  3      PRIM_LCKINDX = SERIAL_FILE (FID);
;  340          1329  3
;  341          1330  3      IF SEARCH_FCB (FID) EQL 0
;  342          1331  3      THEN
;  343          1332  3          ERR_EXIT (SS$_NOSUCHFILE);
;  344          1333  3
;  345          1334  3      HEADER = READ_HEADER (FIB [FIB$W_FID], .FCB);
;  346          1335  2      END;
;  347          1336  2
;  348          1337  2 ! If the file is marked for delete and is not accessed by this user, and
;  349          1338  2 ! the accessor is not the system, deny its existence.
;  350          1339  2 !
;  351          1340  2
;  352          1341  2 IF .CURRENT_WINDOW EQL 0 AND .HEADER[FH2$V_MARKDEL]
;  353          1342  2 AND NOT .BBLOCK [BBLOCK [.PACKET[IRP$L_ARB], ARB$Q_PRIV], PRV$V_BYPASS]
;  354          1343  2 THEN ERR_EXIT (SS$_NOSUCHFILE);
;  355          1344  2
;  356          1345  2 FCB_CREATED = 0;
;  357          1346  2 IF .FCB EQL 0
;  358          1347  2 THEN
;  359          1348  3      BEGIN
;  360          1349  3      FCB_CREATED = 1;
;  361          1350  3      FCB = KERNEL_CALL (CREATE_FCB, .HEADER);
;  362          1351  2      END;
;  363          1352  2 PRIMARY_FCB = .FCB;                        ! record FCB for external use
;  364          1353  2
;  365          1354  2 ! If access is requested, check for conflicts and file protection.
;  366          1355  2 ! then create a window and link everything up.
;  367          1356  2 !
;  368          1357  2
;  369          1358  2 IF .FUNCTION[IO$V_ACCESS]
;  370          1359  2 THEN
;  371          1360  3      BEGIN
;  372          1361  4      IF (.HEADER[FH2$V_LOCKED])
;  373          1362  3      AND NOT .BBLOCK [BBLOCK [.PACKET[IRP$L_ARB], ARB$Q_PRIV], PRV$V_BYPASS]
;  374          1363  3      THEN ERR_EXIT (SS$_FILELOCKED);              ! file is deaccess locked
;  375          1364  3
;  376          1365  4          BEGIN
;  377          1366  4          LOCAL
;  378          1367  4              PREV_MODE;
;  379          1368  4
;  380          1369  4          PREV_MODE = .FCB [FCB$B_ACCLKMODE];
;  381          1370  4
;  382          1371  4          IF NOT ARBITRATE_ACCESS (.FIB [FIB$L_ACCTL], .FCB)
;  383          1372  4          THEN ERR_EXIT (SS$_ACCONFLICT);
;  384          1373  4
;  385          1374  4          CURRENT_WINDOW = CREATE_WINDOW (.FIB[FIB$L_ACCTL], .FIB[FIB$B_WSIZE],
;  386          1375  4                                          .HEADER, .PACKET[IRP$L_PID], .FCB);
;  387          1376  4
;  388          1377  4          IF .CURRENT_WINDOW EQL 0
```

```
;  389          1378  4          THEN
;  390          1379  5              BEGIN
;  391          1380  5
;  392          1381  5              IF .PREV_MODE<0,8> LSSU .FCB [FCB$B_ACCLKMODE]
;  393          1382  5              THEN
;  394          1383  5                  CONV_ACCLOCK (.PREV_MODE, .FCB);
;  395          1384  5
;  396          1385  5              ERR_EXIT (SS$_EXBYTLM);
;  397          1386  4              END;
;  398          1387  4
;  399          1388  4          MAKE_ACCESS (.FCB, .CURRENT_WINDOW, .ABD);
;  400          1389  4
;  401          1390  4          IF .FCB [FCB$V_DELAYTRNC]
;  402          1391  4              AND .FIB [FIB$V_WRITE]
;  403          1392  4          THEN
;  404          1393  5              BEGIN
;  405          1394  5              IF .FCB [FCB$B_ACCLKMODE] LSSU LCK$K_PWMODE
;  406          1395  5              THEN
;  407          1396  5                  IF NOT CONV_ACCLOCK (LCK$K_PWMODE, .FCB)
;  408          1397  5                  THEN
;  409          1398  5                      BUG_CHECK (XQPERR, FATAL, 'Unexpected lock manager error');
;  410          1399  5
;  411          1400  5              FCB [FCB$V_DELAYTRNC] = 0;
;  412          1401  5              FCB [FCB$L_TRUNCVBN] = 0;
;  413          1402  5
;  414          1403  5              CONV_ACCLOCK (.FCB [FCB$B_ACCLKMODE], .FCB);
;  415          1404  4              END;
;  416          1405  4
;  417          1406  3          END;
;  418          1407  3
;  419          1408  3 ! If file expiration is enabled and the volume is writable, check
;  420          1409  3 ! the current expiration date. If it needs to be updated, note this for
;  421          1410  3 ! processing during deaccess. Note that we use the retention parameters
;  422          1411  3 ! from RVN 1 if this is a volume set.
;  423          1412  3 !
;  424          1413  3
;  425          1414  4      CHECK_EXPIRE: BEGIN
;  426          1415  4      PRIMARY_VCB = .CURRENT_VCB;
;  427          1416  4      UCB = .CURRENT_UCB;
;  428          1417  4      IF .PRIMARY_VCB[VCB$W_RVN] NEQ 0
;  429          1418  4      THEN
;  430          1419  5          BEGIN
;  431          1420  5          UCB = .VECTOR [CURRENT_RVT[RVT$L_UCBLST], 0];
;  432          1421  5          IF .UCB EQL 0
;  433          1422  5          THEN LEAVE CHECK_EXPIRE;
;  434          1423  5          PRIMARY_VCB = .UCB[UCB$L_VCB];
;  435          1424  4          END;
;  436          1425  4
;  437          1426  4      IF NOT .BBLOCK [UCB[UCB$L_DEVCHAR], DEV$V_SWL]
;  438          1427  4      AND NOT .FIB[FIB$V_NORECORD]
;  439          1428  4      AND CMPM (2, PRIMARY_VCB[VCB$Q_RETAINMAX], UPLIT (0, 0)) NEQ 0
;  440          1429  4      AND .HEADER[FH2$B_MPOFFSET]-.HEADER[FH2$B_IDOFFSET]
;  441          1430  4          GEQU ($BYTEOFFSET (FI2$Q_EXPDATE) + FI2$S_EXPDATE) / 2
;  442          1431  4      THEN
;  443          1432  5          BEGIN
;  444          1433  5          IDENT_AREA = .HEADER + .HEADER[FH2$B_IDOFFSET]*2;
;  445          1434  5          $GETTIM (TIMADR = DAY_TIME);
```

```
;  446     1435  5          SUBQ (PRIMARY_VCB[VCB$Q_RETAINMIN], DAY_TIME);
;  447     1436  5          IF CMPM (2, IDENT_AREA[FI2$Q_EXPDATE], DAY_TIME) LSS 0
;  448     1437  5          THEN KERNEL_CALL (SET_EXPIRE);
;  449     1438  4          END;
;  450     1439  3      END;                                    ! end of block CHECK_EXPIRE
;  451     1440  3
;  452     1441  2      END;                                    ! end of access processing
;  453     1442  2
;  454     1443  2  ! If the file is multi-header, read the extension headers and create
;  455     1444  2  ! extension FCB's as necessary.
;  456     1445  2  !
;  457     1446  2
;  458     1447  2  IF .FCB_CREATED
;  459     1448  2  THEN
;  460     1449  2      BUILD_EXT_FCBS (.HEADER)
;  461     1450  2  ELSE
;  462     1451  2      IF .FCB [FCB$V_STALE]
;  463     1452  2      THEN
;  464     1453  3          BEGIN
;  465     1454  3
;  466     1455  3          REBLD_PRIM_FCB (.FCB, .HEADER);
;  467     1456  3
;  468     1457  3          BUILD_EXT_FCBS (.HEADER);
;  469     1458  3
;  470     1459  2          END;
;  471     1460  2
;  472     1461  2  !
;  473     1462  2  ! Finally make sure that access is allowed to the file.
;  474     1463  2  !
;  475     1464  2
;  476     1465  2  IF .FUNCTION[IO$V_ACCESS]
;  477     1466  2  THEN
;  478     1467  3      BEGIN
;  479     1468  3      STATUS = CHECK_PROTECT (IF .FIB[FIB$V_EXECUTE]
;  480     1469  3                              AND NOT .FIB[FIB$V_WRITE]
;  481     1470  3                              AND NOT .FIB[FIB$V_NOREAD]
;  482     1471  3                              AND .PACKET[IRP$V_MODE] LEQU 2
;  483     1472  3                              THEN EXEC_ACCESS
;  484     1473  3                              ELSE .FIB[FIB$V_WRITE] OR .FIB[FIB$V_NOREAD],
;  485     1474  3                              .HEADER,
;  486     1475  3                              .FCB,
;  487     1476  3                              MAXU (.PACKET[IRP$V_MODE], .FIB[FIB$B_AGENT_MODE]),
;  488     1477  3                              .FIB[FIB$L_ALT_ACCESS],
;  489     1478  3                              .FIB[FIB$V_ALT_REQ]);
;  490     1479  3
;  491     1480  3      IF .STATUS EQL SS$_NOTALLPRIV
;  492     1481  3      THEN FIB[FIB$V_ALT_GRANTED] = 0;
;  493     1482  3
;  494     1483  3  ! If this is a write access to the index file, the storage map, or the
;  495     1484  3  ! quota file, flush the appropriate cache. Also take out the cache lock
;  496     1485  3  ! if the volume is cluster accessible.
;  497     1486  3  !
;  498     1487  3
;  499     1488  3      IF .CURRENT_WINDOW[WCB$V_WRITE]
;  500     1489  5      AND ((.FIB[FIB$B_FID_NMX] EQL 0
;  501     1490  5            AND .FIB[FIB$W_FID_NUM] LEQU FID$C_BITMAP)
;  502     1491  4           OR .FCB EQL .CURRENT_VCB[VCB$L_QUOTAFCB])
```

```
;  503          1492  3      THEN
;  504          1493  4          BEGIN
;  505          1494  4          ALLOCATION_LOCK ();
;  506          1495  4
;  507          1496  4          IF .FCB EQL .CURRENT_VCB[VCB$L_QUOTAFCB]
;  508          1497  4          THEN FLUSH_QUO_CACHE ()
;  509          1498  4          ELSE IF .FIB[FIB$W_FID_NUM] EQL FID$C_INDEXF
;  510          1499  4          THEN DELETE_FID (0)
;  511          1500  4          ELSE PURGE_EXTENT (0, 0);
;  512          1501  4
;  513          1502  4          RELEASE_LOCKBASIS (-1);
;  514          1503  4          ALLOCATION_UNLOCK ();
;  515          1504  4
;  516          1505  4          IF .BBLOCK [CURRENT_UCB[UCB$L_DEVCHAR2], DEV$V_CLU]
;  517          1506  4          AND .FCB[FCB$L_CACHELKID] EQL 0
;  518          1507  4          THEN
;  519          1508  5              BEGIN
;  520          1509  5              STATUS = CACHE_LOCK (.FCB[FCB$L_LOCKBASIS], FCB[FCB$L_CACHELKID], 2);
;  521          1510  5              IF NOT .STATUS THEN ERR_EXIT (.STATUS);
;  522          1511  4              END;
;  523          1512  3          END;
;  524          1513  2      END;
;  525          1514  2
;  526          1515  2  ! Do read attributes if requested.
;  527          1516  2  !
;  528          1517  2
;  529          1518  2  IF .PACKET[IRP$W_BCNT] GTR ABD$C_ATTRIB
;  530          1519  2  THEN
;  531          1520  3      BEGIN
;  532          1521  3      IF .CURRENT_WINDOW EQL 0
;  533          1522  3      THEN STATUS = CHECK_PROTECT (RDATT_ACCESS, .HEADER, .FCB,
;  534          1523  3                                   MAXU (.PACKET[IRP$V_MODE], .FIB[FIB$B_AGENT_MODE]),
;  535          1524  3                                   .FIB[FIB$L_ALT_ACCESS], .FIB[FIB$V_ALT_REQ]);
;  536          1525  3      IF NOT KERNEL_CALL (READ_ATTRIB, .HEADER, .ABD) THEN ERR_EXIT ();
;  537          1526  3      HEADER = .FILE_HEADER;
;  538          1527  2      END;
;  539          1528  2
;  540          1529  2  IF .STATUS EQL SS$_NOTALLPRIV
;  541          1530  2  THEN FIB[FIB$V_ALT_GRANTED] = 0;
;  542          1531  2
;  543          1532  2  ! If necessary map the file completely.
;  544          1533  2  !
;  545          1534  2
;  546          1535  2  IF .FUNCTION[IO$V_ACCESS]
;  547          1536  2  THEN
;  548          1537  2      IF .CURRENT_WINDOW[WCB$V_CATHEDRAL]
;  549          1538  2      THEN
;  550          1539  2          IF .PRIMARY_FCB[FCB$L_EXFCB] NEQ 0
;  551          1540  2          THEN REMAP_FILE()
;  552          1541  2          ELSE KERNEL_CALL (MARK_COMPLETE, .CURRENT_WINDOW);
;  553          1542  2
;  554          1543  2  RETURN 1;
;  555          1544  2
;  556          1545  1  END;                                   ! end of routine ACCESS

                                                                .TITLE  ACCESS
```

```
                                                .IDENT   \V04-000\

                                                .PSECT   $CODE$,NOWRT,2

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

                                                .EXTRN   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

                               0BFC 00000       .ENTRY   ACCESS, Save R2,R3,R4,R5,R6,R7,R8,R9,R11   ; 1150
             5E              14  C2 00002       SUBL2    #20, SP                                    ;
             58       0C     AA  9E 00005       MOVAB    12(BASE), R8                               ; 1211
             54       90     AA  D0 00009       MOVL     -112(BASE), PACKET                         ; 1249
             7E       20     A4  3C 0000D       MOVZWL   32(PACKET), FUNCTION                       ; 1250
 06          6E              06  E1 00011       BBC      #6, FUNCTION, 1$                           ; 1251
       02    AA     0402     8F  A8 00015       BISW2    #1026, 2(BASE)                             ; 1255
             59       2C     B4  D0 0001B 1$:   MOVL     @44(PACKET), ABD                           ; 1262
                             59  DD 0001F       PUSHL    ABD                                        ; 1263
    0000G    CF              01  FB 00021       CALLS    #1, GET_FIB                                ;
             52              50  D0 00026       MOVL     R0, FIB                                    ;
                      3C     A2  D5 00029       TSTL     60(FIB)                                    ; 1264
                             04  13 0002C       BEQL     2$                                         ;
       38    A2              02  88 0002E       BISB2    #2, 56(FIB)                                ; 1265
 0B          6A              06  E1 00032 2$:   BBC      #6, (BASE), 3$                             ; 1270
                             7E  D4 00036       CLRL     -(SP)                                      ; 1271
                             52  DD 00038       PUSHL    FIB                                        ;
                             59  DD 0003A       PUSHL    ABD                                        ;
    0000G    CF              03  FB 0003C       CALLS    #3, FIND                                   ;
             50       08     AA  D0 00041 3$:   MOVL     8(BASE), R0                                ; 1278
                             13  13 00045       BEQL     5$                                         ;
       04    A2       24     A0  B1 00047       CMPW     36(R0), 4(FIB)                             ; 1280
                             07  12 0004C       BNEQ     4$                                         ;
       08    A2       28     A0  B1 0004E       CMPW     40(R0), 8(FIB)                             ; 1281
                             05  13 00053       BEQL     5$                                         ;
                      08     AA  D4 00055 4$:   CLRL     8(BASE)                                    ; 1284
                             68  D4 00058       CLRL     (R8)                                       ; 1285
 09          6E              06  E0 0005A 5$:   BBS      #6, FUNCTION, 6$                           ; 1292
             05       32     A4  B1 0005E       CMPW     50(PACKET), #5                             ; 1293
                             03  1A 00062       BGTRU    6$                                         ;
                           0331  31 00064       BRW      45$                                        ;
             7E       08     A2  3C 00067 6$:   MOVZWL   8(FIB), -(SP)                              ; 1299
```

```
          0000G   CF         01  FB 0006B         CALLS   #1, SWITCH_VOLUME                          ;
                       04    A2  9F 00070         PUSHAB  4(FIB)                                     ; 1304
          0000G   CF         01  FB 00073         CALLS   #1, SERIAL_FILE                            ;
             18   AA         50  D0 00078         MOVL    R0, 24(BASE)                               ;
                       04    A2  9F 0007C         PUSHAB  4(FIB)                                     ; 1310
          0000G   CF         01  FB 0007F         CALLS   #1, SEARCH_FCB                             ;
                  53         50  D0 00084         MOVL    R0, FCB                                    ;
                       04    AE  D4 00087         CLRL    REALBASIS                                  ; 1311
                       04    AE  9F 0008A         PUSHAB  REALBASIS                                  ; 1312
                             53  DD 0008D         PUSHL   FCB                                        ;
                       04    A2  9F 0008F         PUSHAB  4(FIB)                                     ;
          0000G   CF         03  FB 00092         CALLS   #3, READ_HEADER                            ;
                  56         50  D0 00097         MOVL    R0, HEADER                                 ;
                       04    AE  D5 0009A         TSTL    REALBASIS                                  ; 1314
                             45  13 0009D         BEQL    7$                                         ;
             08   AE   04    AE  B0 0009F         MOVW    REALBASIS, FID                             ; 1320
             0D   AE   06    AE  90 000A4         MOVB    REALBASIS+2, FID+5                         ; 1321
             0C   AE   07    AE  90 000A9         MOVB    REALBASIS+3, FID+4                         ; 1322
                  7E   0C    AE  9A 000AE         MOVZBL  FID+4, -(SP)                               ; 1324
          0000G   CF         01  FB 000B2         CALLS   #1, SWITCH_VOLUME                          ;
                       18    AA  DD 000B7         PUSHL   24(BASE)                                   ; 1326
          0000G   CF         01  FB 000BA         CALLS   #1, RELEASE_SERIAL_LOCK                    ;
                       08    AE  9F 000BF         PUSHAB  FID                                        ; 1328
          0000G   CF         01  FB 000C2         CALLS   #1, SERIAL_FILE                            ;
             18   AA         50  D0 000C7         MOVL    R0, 24(BASE)                               ;
                       08    AE  9F 000CB         PUSHAB  FID                                        ; 1330
          0000G   CF         01  FB 000CE         CALLS   #1, SEARCH_FCB                             ;
                             50  D5 000D3         TSTL    R0                                         ;
                             1B  13 000D5         BEQL    8$                                         ;
                             53  DD 000D7         PUSHL   FCB                                        ; 1334
                       04    A2  9F 000D9         PUSHAB  4(FIB)                                     ;
          0000G   CF         02  FB 000DC         CALLS   #2, READ_HEADER                            ;
                  56         50  D0 000E1         MOVL    R0, HEADER                                 ;
                             68  D5 000E4 7$:     TSTL    (R8)                                       ; 1341
                             0F  12 000E6         BNEQ    9$                                         ;
                       35    A6  95 000E8         TSTB    53(HEADER)                                 ;
                             0A  18 000EB         BGEQ    9$                                         ;
   05        58   B4         1D  E0 000ED         BBS     #29, @88(PACKET), 9$                       ; 1342
                     0910    8F  BF 000F2 8$:     CHMU    #2320                                      ; 1343
                                 04 000F6         RET                                                ;
                             5B  D4 000F7 9$:     CLRL    FCB_CREATED                                ; 1345
                             53  D5 000F9         TSTL    FCB                                        ; 1346
                             0D  12 000FB         BNEQ    10$                                        ;
                  5B         01  D0 000FD         MOVL    #1, FCB_CREATED                            ; 1349
                             56  DD 00100         PUSHL   HEADER                                     ; 1350
          0000G   CF         01  FB 00102         CALLS   #1, CREATE_FCB                             ;
                  53         50  D0 00107         MOVL    R0, FCB                                    ;
             08   AA         53  D0 0010A 10$:    MOVL    FCB, 8(BASE)                               ; 1352
   03             6E         06  E0 0010E         BBS     #6, FUNCTION, 11$                          ; 1358
                           011E  31 00112         BRW     25$                                        ;
   0A        34   A6         06  E1 00115 11$:    BBC     #6, 52(HEADER), 12$                        ; 1361
   05        58   B4         1D  E0 0011A         BBS     #29, @88(PACKET), 12$                      ; 1362
                     08A8    8F  BF 0011F         CHMU    #2216                                      ; 1363
                                 04 00123         RET                                                ;
                  55   0B    A3  9A 00124 12$:    MOVZBL  11(FCB), PREV_MODE                         ; 1369
                  51         53  D0 00128         MOVL    FCB, R1                                    ; 1371
                             62  D0 0012B         MOVL    (FIB), R0                                  ;
```

```
                        0000G 30 0012E          BSBW    ARBITRATE_ACCESS
                    05    50 E8 00131          BLBS    R0, 13$
                0800    8F BF 00134          CHMU    #2048                                   ;1372
                           04 00138          RET
                        53 DD 00139 13$:     PUSHL   FCB                                     ;1375
                     0C    A4 DD 0013B          PUSHL   12(PACKET)
                        56 DD 0013E          PUSHL   HEADER
                    7E    03 A2 98 00140          CVTBL   3(FIB), -(SP)                    ;1374
                        62 DD 00144          PUSHL   (FIB)
                0000G   CF    05 FB 00146          CALLS   #5, CREATE_WINDOW
                    68        50 D0 0014B          MOVL    R0, (R8)
                        68 D5 0014E          TSTL    (R8)                                    ;1377
                        14 12 00150          BNEQ    15$
                  0B    A3    55 91 00152          CMPB    PREV_MODE, 11(FCB)              ;1381
                        09 1E 00156          BGEQU   14$
                        53 DD 00158          PUSHL   FCB                                     ;1383
                        55 DD 0015A          PUSHL   PREV_MODE
                0000G   CF    02 FB 0015C          CALLS   #2, CONV_ACCLOCK
                2A14    8F BF 00161 14$:     CHMU    #10772                                  ;1385
                           04 00165          RET
                        59 DD 00166 15$:     PUSHL   ABD                                     ;1388
                        68 DD 00168          PUSHL   (R8)
                        53 DD 0016A          PUSHL   FCB
            00000000G   00    03 FB 0016C          CALLS   #3, MAKE_ACCESS
        2C      23  A3        01 E1 00173          BBC     #1, 35(FCB), 17$                ;1390
                    28    01  A2 E9 00178          BLBC    1(FIB), 17$                     ;1391
                    04    0B  A3 91 0017C          CMPB    11(FCB), #4                     ;1394
                        10 1E 00180          BGEQU   16$
                        53 DD 00182          PUSHL   FCB                                     ;1396
                        04 DD 00184          PUSHL   #4
                0000G   CF    02 FB 00186          CALLS   #2, CONV_ACCLOCK
                    04        50 E8 00188          BLBS    R0, 16$
                           FEFF 0018E          BUGW                                            ;1398
                          0000* 00190          .WORD   <BUG$_XQPERR!4>
                  23    A3    02 8A 00192 16$:     BICB2   #2, 35(FCB)                     ;1400
                        50  A3 D4 00196          CLRL    80(FCB)                         ;1401
                        53 DD 00199          PUSHL   FCB                                     ;1403
                    7E    0B  A3 9A 0019B          MOVZBL  11(FCB), -(SP)
                0000G   CF    02 FB 0019F          CALLS   #2, CONV_ACCLOCK
                        55    98 AA D0 001A4 17$:     MOVL    -104(BASE), PRIMARY_VCB         ;1415
                        50    94 AA D0 001A8          MOVL    -108(BASE), UCB                 ;1416
                              0E A5 B5 001AC          TSTW    14(PRIMARY_VCB)                 ;1417
                              0E 13 001AF          BEQL    18$
                        51    9C AA D0 001B1          MOVL    -100(BASE), R1                  ;1420
                        50    44 A1 D0 001B5          MOVL    68(R1), UCB
                              78 13 001B9          BEQL    25$                             ;1421
                        55    34 A0 D0 001BB          MOVL    52(UCB), PRIMARY_VCB            ;1423
    6F      3B  A0              01 E0 001BF 18$:     BBS     #1, 59(UCB), 25$                ;1426
    6B          62              15 E0 001C4          BBS     #21, (FIB), 25$                 ;1427
                50              01 CE 001C8          MNEGL   #1, R0                          ;1428
                FE2B  CF    78  A5 D1 001CB          CMPL    120(PRIMARY_VCB), P.AAA+4
                              10 19 001D1          BLSS    21$
                              0A 14 001D3          BGTR    19$
                FE1D  CF    74  A5 D1 001D5          CMPL    116(PRIMARY_VCB), P.AAA
                              04 13 001DB          BEQL    20$
                              04 1F 001DD          BLSSU   21$
                              50 D6 001DF 19$:     INCL    R0
```

```
                                           50  D6 001E1 20$:    INCL    R0                                    ;
                                           50  D5 001E3 21$:    TSTL    R0                                    ;
                                           4C  13 001E5         BEQL    25$                                   ;
                            50       01    A6  9A 001E7         MOVZBL  1(HEADER), R0                         ; 1429
                            51             66  9A 001EB         MOVZBL  (HEADER), R1                          ;
                            50             51  C2 001EE         SUBL2   R1, R0                                ;
                            17             50  D1 001F1         CMPL    R0, #23                               ; 1430
                                           3D  1F 001F4         BLSSU   25$                                   ;
                            50             66  9A 001F6         MOVZBL  (HEADER), R0                          ; 1433
                            57           6640  3E 001F9         MOVAW   (HEADER)[R0], IDENT_AREA              ;
                                     10    AE  9F 001FD         PUSHAB  DAY_TIME                              ; 1434
                 00000000G  00             01  FB 00200         CALLS   #1, SYS$GETTIM                        ;
                        10  AE       6C    A5  C2 00207         SUBL2   108(PRIMARY_VCB), DAY_TIME            ; 1435
                        14  AE       70    A5  D9 0020C         SBWC    112(PRIMARY_VCB), DAY_TIME            ;
                            50             01  CE 00211         MNEGL   #1, R0                                ; 1436
                        14  AE       2A    A7  D1 00214         CMPL    42(IDENT_AREA), DAY_TIME              ;
                                           0F  19 00219         BLSS    24$                                   ;
                                           09  14 0021B         BGTR    22$                                   ;
                        10  AE       26    A7  D1 0021D         CMPL    38(IDENT_AREA), DAY_TIME              ;
                                           04  13 00222         BEQL    23$                                   ;
                                           04  1F 00224         BLSSU   24$                                   ;
                                           50  D6 00226 22$:    INCL    R0                                    ;
                                           50  D6 00228 23$:    INCL    R0                                    ;
                                           50  D5 0022A 24$:    TSTL    R0                                    ;
                                           05  18 0022C         BGEQ    25$                                   ;
                     0000V  CF             00  FB 0022E         CALLS   #0, SET_EXPIRE                        ; 1437
                            0D             5B  E8 00233 25$:    BLBS    FCB_CREATED, 26$                      ; 1447
                            10       23    A3  E9 00236         BLBC    35(FCB), 27$                          ; 1451
                                   0048    8F  BB 0023A         PUSHR   #^M<R3,R6>                            ; 1455
                     0000G  CF             02  FB 0023E         CALLS   #2, REBLD_PRIM_FCB                    ;
                                           56  DD 00243 26$:    PUSHL   HEADER                                ; 1457
                     0000G  CF             01  FB 00245         CALLS   #1, BUILD_EXT_FCBS                    ;
                03          6E             06  E0 0024A 27$:    BBS     #6, FUNCTION, 28$                     ; 1465
                                         00D3  31 0024E         BRW     38$                                   ;
  7E        38  A2              01         00  EF 00251 28$:    EXTZV   #0, #1, 56(FIB), -(SP)                ; 1478
                                     3C    A2  DD 00257         PUSHL   60(FIB)                               ; 1477
  7E        0B  A4              02         00  EF 0025A         EXTZV   #0, #2, 11(PACKET), -(SP)             ; 1476
                                6E   2E    A2  91 00260         CMPB    46(FIB), (SP)                         ;
                                           04  1B 00264         BLEQU   29$                                   ;
                                6E   2E    A2  9A 00266         MOVZBL  46(FIB), (SP)                         ;
                                           53  DD 0026A 29$:    PUSHL   FCB                                   ; 1475
                                           56  DD 0026C         PUSHL   HEADER                                ; 1474
                                14   02    A2  E9 0026E         BLBC    2(FIB), 30$                           ; 1468
                                10   01    A2  E8 00272         BLBS    1(FIB), 30$                           ; 1469
                0C              62         0A  E0 00276         BBS     #10, (FIB), 30$                       ; 1470
  02        0B  A4              02         00  ED 0027A         CMPZV   #0, #2, 11(PACKET), #2                ; 1471
                                           04  1A 00280         BGTRU   30$                                   ;
                                           06  DD 00282         PUSHL   #6                                    ; 1468
                                           10  11 00284         BRB     31$                                   ;
  50        01  A2              01         00  EF 00286 30$:    EXTZV   #0, #1, 1(FIB), R0                    ; 1473
  51            62              01         0A  EF 0028C         EXTZV   #10, #1, (FIB), R1                    ;
                                50         51  C8 00291         BISL2   R1, R0                                ;
                                           50  DD 00294         PUSHL   R0                                    ;
                     0000G  CF             06  FB 00296 31$:    CALLS   #6, CHECK_PROTECT                     ; 1468
                            57             50  D0 0029B         MOVL    R0, STATUS                            ;
                 00000681   8F             57  D1 0029E         CMPL    STATUS, #1665                         ; 1480
                                           04  12 002A5         BNEQ    32$                                   ;
```

```
               38  A2        02  8A 002A7       BICB2  #2, 56(FIB)                        ; 1481
                   50        68  D0 002AB 32$:  MOVL   (R8), R0                           ; 1488
          71       0B  A0    01  E1 002AE       BBC    #1, 11(R0), 38$
                         09  A2  95 002B3       TSTB   9(FIB)                             ; 1489
                             06  12 002B6       BNEQ   33$
                   02    04  A2  B1 002B8       CMPW   4(FIB), #2                         ; 1490
                             0A  1B 002BC       BLEQU  34$
                   50    98  AA  D0 002BE 33$:  MOVL   -104(BASE), R0                     ; 1491
               54  A0        53  D1 002C2       CMPL   FCB, 84(R0)
                             5C  12 002C6       BNEQ   38$
            0000G  CF        00  FB 002C8 34$:  CALLS  #0, ALLOCATION_LOCK                ; 1494
                   50    98  AA  D0 002CD       MOVL   -104(BASE), R0                     ; 1496
               54  A0        53  D1 002D1       CMPL   FCB, 84(R0)
                             07  12 002D5       BNEQ   35$
            0000G  CF        00  FB 002D7       CALLS  #0, FLUSH_QUO_CACHE                ; 1497
                             16  11 002DC       BRB    37$
                   01    04  A2  B1 002DE 35$:  CMPW   4(FIB), #1                         ; 1498
                             09  12 002E2       BNEQ   36$
                             7E  D4 002E4       CLRL   -(SP)                              ; 1499
            0000G  CF        01  FB 002E6       CALLS  #1, DELETE_FID
                             07  11 002EB       BRB    37$
                             7E  7C 002ED 36$:  CLRQ   -(SP)                              ; 1500
            0000G  CF        02  FB 002EF       CALLS  #2, PURGE_EXTENT
                   7E        01  CE 002F4 37$:  MNEGL  #1, -(SP)                          ; 1502
            0000G  CF        01  FB 002F7       CALLS  #1, RELEASE_LOCKBASIS
            0000G  CF        00  FB 002FC       CALLS  #0, ALLOCATION_UNLOCK              ; 1503
                   50    94  AA  D0 00301       MOVL   -108(BASE), R0                     ; 1505
                   1B    3C  A0  E9 00305       BLBC   60(R0), 38$
                         54  A3  D5 00309       TSTL   84(FCB)                            ; 1506
                             16  12 0030C       BNEQ   38$
                             02  DD 0030E       PUSHL  #2                                 ; 1509
                         54  A3  9F 00310       PUSHAB 84(FCB)
                         4C  A3  DD 00313       PUSHL  76(FCB)
            0000G  CF        03  FB 00316       CALLS  #3, CACHE_LOCK
                   57        50  D0 0031B       MOVL   R0, STATUS
                   03        57  E8 0031E       BLBS   STATUS, 38$                        ; 1510
                             57  BF 00321       CHMU   STATUS
                                 04 00323       RET
                   05    32  A4  B1 00324 38$:  CMPW   50(PACKET), #5                     ; 1518
                             3E  1B 00328       BLEQU  42$
                             68  D5 0032A       TSTL   (R8)                               ; 1521
                             27  12 0032C       BNEQ   40$
  7E       38  A2        01  00  EF 0032E       EXTZV  #0, #1, 56(FIB), -(SP)             ; 1524
                         3C  A2  DD 00334       PUSHL  60(FIB)
  7E       0B  A4        02  00  EF 00337       EXTZV  #0, #2, 11(PACKET), -(SP)          ; 1523
                   6E    2E  A2  91 0033D       CMPB   46(FIB), (SP)
                             04  1B 00341       BLEQU  39$
                   6E    2E  A2  9A 00343       MOVZBL 46(FIB), (SP)
                             53  DD 00347 39$:  PUSHL  FCB                                ; 1522
                             56  DD 00349       PUSHL  HEADER
                             04  DD 0034B       PUSHL  #4
            0000G  CF        06  FB 0034D       CALLS  #6, CHECK_PROTECT
                   57        50  D0 00352       MOVL   R0, STATUS
                       0240  8F  BB 00355 40$:  PUSHR  #^M<R6,R9>                         ; 1525
            0000G  CF        02  FB 00359       CALLS  #2, READ_ATTRIB
                   03        50  E8 0035E       BLBS   R0, 41$
                             00  BF 00361       CHMU   #0
```

```
                                       04 00363          RET                                          ;
                56      04    AA       D0 00364 41$:     MOVL    4(BASE), HEADER                      ; 1526
      00000681  8F            57       D1 00368 42$:     CMPL    STATUS, #1665                        ; 1529
                              04       12 0036F          BNEQ    43$                                  ;
                38    A2      02       8A 00371          BICB2   #2, 56(FIB)                          ; 1530
  1F                  6E      06       E1 00375 43$:     BBC     #6, FUNCTION, 45$                    ; 1535
                      51      68       D0 00379          MOVL    (R8), R1                             ; 1537
  17            0B    A1      06       E1 0037C          BBC     #6, 11(R1), 45$                      ;
                      50      08 AA    D0 00381          MOVL    8(BASE), R0                          ; 1539
                              0C A0    D5 00385          TSTL    12(R0)                               ;
                              07       13 00388          BEQL    44$                                  ;
           0000G  CF          00       FB 0038A          CALLS   #0, REMAP_FILE                       ; 1540
                              07       11 0038F          BRB     45$                                  ;
                              51       DD 00391 44$:     PUSHL   R1                                   ; 1541
           0000G  CF          01       FB 00393          CALLS   #1, MARK_COMPLETE                    ;
                      50      01       D0 00398 45$:     MOVL    #1, R0                               ; 1543
                                       04 0039B          RET                                          ; 1545

; Routine Size:  924 bytes,    Routine Base:  $CODE$ + 0008
```

```
;    1 GLOBAL ROUTINE SET_EXPIRE : L_NORM =
;    1
;    1 !++
;    1 !
;    1 ! FUNCTIONAL DESCRIPTION:
;    1 !
;    1 !       This routine sets the bit in a window marking the file for
;    1 !       expiration date recording when it is closed.
;    1 !
;    1 ! CALLING SEQUENCE:
;    1 !       SET_EXPIRE ()
;    1 !
;    1 ! INPUT PARAMETERS:
;    1 !       NONE
;    1 !
;    1 ! IMPLICIT INPUTS:
;    1 !       CURRENT_WINDOW: address of file window
;    1 !
;    1 ! OUTPUT PARAMETERS:
;    1 !       NONE
;    1 !
;    1 ! IMPLICIT OUTPUTS:
;    1 !       NONE
;    1 !
;    1 ! ROUTINE VALUE:
;    1 !       1
;    1 !
;    1 ! SIDE EFFECTS:
;    1 !       expire bit set
;    1 !
;    1 !--
;    1
;    2 BEGIN
;    2
;    2 BIND_COMMON;
;    2
;    2 CURRENT_WINDOW[WCB$V_EXPIRE] = 1;
;    2
;    2 1
;    1 END;                                  ! End of routine SET_EXPIRE



                               0000 00000         .ENTRY  SET_EXPIRE, Save nothing      ; 1546
                50       0C    AA  D0 00002         MOVL    12(BASE), R0                  ; 1582
        0B      A0       80    8F  88 00006         BISB2   #128, 11(R0)                  ;
                50             01  D0 0000B         MOVL    #1, R0                        ; 1585
                                   04 0000E         RET                                   ;

; Routine Size:  15 bytes,    Routine Base:  $CODE$ + 03A4


;    1
;    1 END
;    0 ELUDOM
```

```
;                                PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $CODE$                         947  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                              Library Statistics

;                                        --------  Symbols --------      Pages       Processing
;       File                             Total     Loaded   Percent      Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619         86         0        1000        00:02.0



;                                   COMMAND QUALIFIERS

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

; Size:         939 code + 8 data bytes
; Run Time:        00:40.5
; Elapsed Time:    01:32.8
; Lines/CPU Min:    2350
; Lexemes/CPU-Min: 42153
; Memory Used:  425 pages
; Compilation Complete
```

FCPDEF
B32

ACLCNTRL
LIS

ACLSUBR
LIS

SNDERL
LIS

ACPCNTRL
LIS

F11X

F11BXQP
MAP

ACCESS
LIS

TRUNC
LIS

FILESERV
MAP

WITURN
LIS

SNDSMB
LIS