G 16

```
FFFFFFFFFFFFFFF  DDDDDDDDDDDD     LLL
FFFFFFFFFFFFFFF  DDDDDDDDDDDD     LLL
FFFFFFFFFFFFFFF  DDDDDDDDDDDD     LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFFFFFFFFFFF     DDD         DDD  LLL
FFFFFFFFFFFF     DDD         DDD  LLL
FFFFFFFFFFFF     DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDD         DDD  LLL
FFF              DDDDDDDDDDDD     LLLLLLLLLLLLLLL
FFF              DDDDDDDDDDDD     LLLLLLLLLLLLLLL
FFF              DDDDDDDDDDDD     LLLLLLLLLLLLLLL
```

```
**FILE**ID**FDLPARSE                                    B  5

FFFFFFFFFF  DDDDDDDD    LL          PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
FFFFFFFFFF  DDDDDDDD    LL          PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
FF          DD      DD  LL          PP      PP  AA      AA  RR      RR  SS          EE
FF          DD      DD  LL          PP      PP  AA      AA  RR      RR  SS          EE
FF          DD      DD  LL          PP      PP  AA      AA  RR      RR  SS          EE
FF          DD      DD  LL          PP      PP  AA      AA  RR      RR  SS          EE
FFFFFFFF    DD      DD  LL          PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
FFFFFFFF    DD      DD  LL          PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
FF          DD      DD  LL          PP          AAAAAAAAAA  RR  RR              SS  EE
FF          DD      DD  LL          PP          AAAAAAAAAA  RR  RR              SS  EE
FF          DD      DD  LL          PP          AA      AA  RR    RR            SS  EE
FF          DD      DD  LL          PP          AA      AA  RR    RR            SS  EE              ....
FF          DDDDDDDD    LLLLLLLLLL  PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....
FF          DDDDDDDD    LLLLLLLLLL  PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....
                                                                                                    ....

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
```

```
FDL
V04

; R
```

```
0 %TITLE  'FDL$PARSE'
0 %SBTTL  'FDL Parse Action Routines'
0 MODULE  FDLPARSE          ( IDENT='V04-000',
0                           ADDRESSING_MODE ( EXTERNAL = GENERAL ),
0                           ADDRESSING_MODE ( NONEXTERNAL = GENERAL ),
0                           OPTLEVEL=3
0                           ) =
0
1 BEGIN
1
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
```

```
1 !++
1 !
1 ! Facility:
1 !                     RMS-32 FDL Utilities
1 !
1 ! Environment:
1 !                     VAX/VMS Operating System
1 !
1 ! Abstract:
1 !                     Routines which fill the rms control blocks
1 !                     for the FDL parser
1 !
1 ! Contents:
1 !                     INIT_PARSE
1 !                     LINE_PARSED
1 !                     SET_AREA_P
1 !                     SET_DATE_P
1 !                     SET_JNL_P
1 !                     SET_ACL_P
1 !                     SET_FILE_P
1 !                     SET_KEY_P
1 !                     SET_RECORD_P
1 !                     SET_ACCESS_P
1 !                     SET_SHARING_P
1 !                     SET_CONNECT_P
1 !                     SET_PROT
1 !                     ALLOCATE_XAB
1 !                     FIND_ID
1 !                     GET_VM
1 !                     FREE_VM
1 !
1 !--
```

```
1 !
1 ! Author:          Keith B Thompson          Creation date:  July-1981
1 !
1 !
1 ! Modified by:
1 !
1 !           V03-011 RRB0015          Rowland R. Bradley        29 Feb 1984
1 !                   Comment out references to ERASE_ON_DELETE and ACL support.
1 !                   Not supported for V4.0.
1 !
1 !           V03-010 RRB0008          Rowland R. Bradley        19 Jan 1984
1 !                   Support NULL strings in file name.
1 !
1 !           V03-009 KFH0007          Ken Henderson             10 Sep 1983
1 !                   Support for named UICs
1 !
1 !           V03-008 KFH0006          Ken Henderson             29 Jul 1983
1 !                   Check status of call to LIB$...
1 !                   Added DEFERRED_WRITE, ERASE_ON_DELETE
1 !
1 !           V03-007 KFH0005          Ken Henderson             6 Jan 1983
1 !                   Fixed allocation of keyname buffer
1 !
1 !           V03-006 KFH0004          Ken Henderson             21 Dec 1982
1 !                   Deleted unused ref to tpa_block
1 !
1 !           V03-005 KFH0003          Ken Henderson             22 Nov 1982
1 !                   Add support for default and main
1 !                   parses in FDL$PARSE
1 !                   Fix FDL$$FREE_VM to signal status
1 !
1 !           V03-004 KFH0002          Ken Henderson             6-Oct-1982
1 !                   Add support for Journal, Access,
1 !                   ACL, Sharing, Connect primaries
1 !
1 !           V03-003 KBT0069          Keith B. Thompson         24-Jun-1982
1 !                   Initialize the length in fdl$ab_item
1 !
1 !           V03-002 KBT0030          Keith Thompson            30-Mar-1982
1 !                   Fix error processing of the date & time stuff
1 !
1 !           V03-001 KFH0001          Ken Henderson   29 March 1982
1 !                   Fixed SET_AREA_P to set LBN
1 !                   instead of VBN for volume placement
1 !
1 !****
```

```
1
1 PSECT
1               OWN     = _FDL$OWN          (PIC),
1               GLOBAL  = _FDL$GLOBAL       (PIC),
1               PLIT    = _FDL$PLIT         (SHARE,PIC),
1               CODE    = _FDL$CODE         (SHARE,PIC);
1
1 LIBRARY 'SYS$LIBRARY:STARLET';
1 REQUIRE 'SRC$:FDLUTIL';
1 REQUIRE 'LIB$:FDLPARDEF';
1
1 EXTERNAL ROUTINE
1               LIB$GET_VM,
1               LIB$FREE_VM,
1               FDL$$RMS_ERROR              : NOVALUE;
1
1 DEFINE_ERROR_CODES;
1
1 FORWARD ROUTINE
1               SET_AREA_P          : NOVALUE,
1               SET_DATE_P          : NOVALUE,
1               SET_JNL_P           : NOVALUE,
1               SET_ACL_P           : NOVALUE,
1               SET_FILE_P          : NOVALUE,
1               SET_KEY_P           : NOVALUE,
1               SET_RECORD_P        : NOVALUE,
1               SET_ACCESS_P        : NOVALUE,
1               SET_SHARING_P       : NOVALUE,
1               SET_CONNECT_P       : NOVALUE,
1               SET_PROT            : NOVALUE,
1               ALLOCATE_XAB,
1               FIND_ID             : NOVALUE,
1               FDL$$GET_VM,
1               FDL$$FREE_VM        : NOVALUE;
1
1 EXTERNAL
1               FDL$AB_TPARSE_BLOCK         : BLOCK [ ,BYTE ],
1               FDL$AB_ITEM                 : DESC_BLK,
1               FDL$AB_CTRL                 : BLOCK [ ,BYTE ],
1               FDL$GL_PCALL,
1               FDL$GL_STMNTNUM,
1               FDL$GL_PRIMARY,
1               FDL$GL_PRINUM,
1               FDL$AB_PRICTRL,
1               FDL$GL_SECONDARY,
1               FDL$GL_SECNUM,
1               FDL$GL_QUALIFIER,
1               FDL$GL_NUMBER,
1               FDL$GL_SWITCH,
1               FDL$GL_OWNER_UIC,
1               FDL$GL_SPARET,
1               FDL$GL_PROTECTION,
1               FDL$GL_FID1,
1               FDL$GL_FID2,
1               FDL$GL_FID3,
1               FDL$AB_AREA_BKZ             : REF VECTOR [ ,BYTE ],
;               FDL$AL_DATE_TIME            : VECTOR [,LONG ],
1               FDL$AB_STRING               : DESC_BLK,
1
1               FDL$AB_PARSED_FAB           : REF $FAB_DECL,
1               FDL$AB_PARSED_RAB           : REF $RAB_DECL;
1
1 LITERAL
1               SPACE    = 32;
1
1 OWN
1               HIGHEST_AREA_NO : BYTE,
1               CURRENT_XAB     : REF BLOCK [ ,BYTE ],
1               END_XAB         : REF BLOCK [ ,BYTE ],
1
1               JNL_XAB         : REF $XABJNL_DECL,      ! Journal XAB
1               DATE_XAB        : REF $XABDAT_DECL,      ! Date XAB
1               REVISION_XAB    : REF $XABRDT_DECL,      ! Revision Date and Time XAB
1               PROTECTION_XAB  : REF $XABPRO_DECL;      ! Protection XAB
1
```

```
1 %SBTTL  'INIT PARSE'
1 GLOBAL ROUTINE  FDL$$INIT_PARSE : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Init variables and allocate a buffer for the area bucket sizes
1 !
1 ! Calling Sequence:
1 !
1 !       fdl$$init_parse()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routines Called:
1 !
1 !       lib$get_vm
1 !
1 ! Side Effects:
1 !
1 !       Allocates a buffer pointed to by FDL$AB_AREA_BKZ
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         BYTES;
2
2     ! Set the parse control bits
2     !
2     FDL$AB_CTRL [ FDL$V_STATUS ] = _SET;
2     FDL$AB_CTRL [ FDL$V_INITIAL ] = _SET;
2
2     ! Clear the other CTRL bits except the following ones:
2     !   PCALL
2     !   DCL
2     !   STRING_SPEC
2     !   GCALL
2     !
2     FDL$AB_CTRL [ FDL$V_WARNING ] = _CLEAR;
2     FDL$AB_CTRL [ FDL$V_PRIMARY ] = _CLEAR;
2     FDL$AB_CTRL [ FDL$V_NEWPRI ] = _CLEAR;
2     FDL$AB_CTRL [ FDL$V_SECONDARY ] = _CLEAR;
```

```
;  247       0965  2     FDL$AB_CTRL [ FDL$V_COMMENT ] = _CLEAR;
;  248       0966  2     FDL$AB_CTRL [ FDL$V_LINECMT ] = _CLEAR;
;  249       0967  2     FDL$AB_CTRL [ FDL$V_APOST_PRES ] = _CLEAR;
;  250       0968  2     FDL$AB_CTRL [ FDL$V_QUOTE_PRES ] = _CLEAR;
;  251       0969  2     FDL$AB_CTRL [ FDL$V_USED_STRING ] = _CLEAR;
;  252       0970  2
;  253       0971  2     ! Initialize the item length for fdl$get_line
;  254       0972  2     !
;  255       0973  2     FDL$AB_ITEM [ DSC$W_LENGTH ] = 0;
;  256       0974  2
;  257       0975  2     IF NOT .FDL$AB_CTRL [ FDL$V_REPARSE ]
;  258       0976  2     THEN
;  259       0977  3         BEGIN
;  260       0978  3
;  261       0979  3         ! Clear the pointers to xabs
;  262       0980  3         !
;  263       0981  3         JNL_XAB          = _CLEAR;
;  264       0982  3         DATE_XAB         = _CLEAR;
;  265       0983  3         REVISION_XAB     = _CLEAR;
;  266       0984  3         PROTECTION_XAB   = _CLEAR;
;  267       0985  3
;  268       0986  2         END;
;  269       0987  2
;  270       0988  2     ! Clear misc
;  271       0989  2     !
;  272       0990  2     FDL$GL_STMNTNUM      = 0;
;  273       0991  2     FDL$AB_PRICTRL       = _CLEAR;
;  274       0992  2     CURRENT_XAB          = _CLEAR;
;  275       0993  2     HIGHEST_AREA_NO      = 0;
;  276       0994  2
;  277       0995  2     ! Allocate memory for the area bucket size array NOTE: Use lib$get_vm so
;  278       0996  2     ! we can return this in fdl$$finish_parse
;  279       0997  2     !
;  280       0998  2     BYTES = 256;
;  281       0999  2
;  282       1000  2     IF NOT LIB$GET_VM ( BYTES,FDL$AB_AREA_BKZ )
;  283       1001  2     THEN
;  284       1002  2         SIGNAL_STOP ( FDL$_INSVIRMEM );
;  285       1003  2
;  286       1004  2     ! Zero the values
;  287       1005  2     !
;  288       1006  2     CH$FILL( 0,.BYTES,.FDL$AB_AREA_BKZ );
;  289       1007  2
;  290       1008  2     RETURN
;  291       1009  2
;  292       1010  1     END;


                                                       .TITLE  FDLPARSE VAX-11 FDL Utilities
                                                       .IDENT  \V04-000\

                                                       .PSECT  _FDL$OWN,NOEXE, PIC,2

                                            00000 HIGHEST_AREA_NO:
                                                       .BLKB   1
                                            00001      .BLKB   3
                                            00004 CURRENT_XAB:
```

```
                                                           .BLKB   4
                                   00008 END_XAB:.BLKB   4
                                   0000C JNL_XAB:.BLKB   4
                                   00010 DATE_XAB:
                                                           .BLKB   4
                                   00014 REVISION_XAB:
                                                           .BLKB   4
                                   00018 PROTECTION_XAB:
                                                           .BLKB   4

                                                           .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                           .EXTRN  FDL$$RMS_ERROR, FDL$_FACILITY
                                                           .EXTRN  FDL$_FAO_MAX, FDL$_ABKW
                                                           .EXTRN  FDL$_ABPRIKW, FDL$_CREATE
                                                           .EXTRN  FDL$_CREATED, FDL$_CREATEDSTM
                                                           .EXTRN  FDL$_FDLERROR, FDL$_ILL_ARG
                                                           .EXTRN  FDL$_INSVIRMEM, FDL$_INVBLK
                                                           .EXTRN  FDL$_INVDATIM, FDL$_MULPRI
                                                           .EXTRN  FDL$_MULSEC, FDL$_NOQUAL
                                                           .EXTRN  FDL$_NULLPRI, FDL$_OPENFDL
                                                           .EXTRN  FDL$_OUTORDER, FDL$_OPENOUT
                                                           .EXTRN  FDL$_WRITEERR, FDL$_READERR
                                                           .EXTRN  FDL$_RFLOC, FDL$_TITLE
                                                           .EXTRN  FDL$_SYNTAX, FDL$_VALPRI
                                                           .EXTRN  FDL$_UNQUAKW, FDL$_UNPRIKW
                                                           .EXTRN  FDL$_UNSECKW, FDL$_WARNING
                                                           .EXTRN  FDL$AB_TPARSE_BLOCK
                                                           .EXTRN  FDL$AB_ITEM, FDL$AB_CTRL
                                                           .EXTRN  FDL$GL_PCALL, FDL$GL_STMNTNUM
                                                           .EXTRN  FDL$GL_PRIMARY, FDL$GL_PRINUM
                                                           .EXTRN  FDL$AB_PRICTRL, FDL$GL_SECONDARY
                                                           .EXTRN  FDL$GL_SECNUM, FDL$GL_QUALIFIER
                                                           .EXTRN  FDL$GL_NUMBER, FDL$GL_SWITCH
                                                           .EXTRN  FDL$GL_OWNER_UIC
                                                           .EXTRN  FDL$GL_SPARET, FDL$GL_PROTECTION
                                                           .EXTRN  FDL$GL_FID1, FDL$GL_FID2
                                                           .EXTRN  FDL$GL_FID3, FDL$AB_AREA_BKZ
                                                           .EXTRN  FDL$AL_DATE_TIME
                                                           .EXTRN  FDL$AB_STRING, FDL$AB_PARSED_FAB
                                                           .EXTRN  FDL$AB_PARSED_RAB

                                                           .PSECT  _FDL$CODE,NOWRT, SHR, PIC,2

                                      01FC 00000           .ENTRY  FDL$$INIT_PARSE, Save R2,R3,R4,R5,R6,R7,R8  ; 0909
                    58 00000000G   00   9E 00002           MOVAB   FDL$AB_AREA_BKZ, R8                          ;
                    57 00000000G   00   9E 00009           MOVAB   FDL$AB_CTRL, R7                              ;
                    56 00000000'   00   9E 00010           MOVAB   JNL_XAB, R6                                  ;
                    5E             04   C2 00017           SUBL2   #4, SP                                       ;
  67          03    00             01   F0 0001A           INSV    #1, #0, #3, FDL$AB_CTRL                      ; 0952
                    67       80    8F   88 0001F           BISB2   #128, FDL$AB_CTRL                            ; 0953
                    67     E378    8F   AA 00023           BICW2   #58232, FDL$AB_CTRL                          ; 0969
                       00000000G   00   B4 00028           CLRW    FDL$AB_ITEM                                  ; 0973
                    05       02    A7   E8 0002E           BLBS    FDL$AB_CTRL+2, 1$                            ; 0975
                                   66   7C 00032           CLRQ    JNL_XAB                                      ; 0981
                             08    A6   7C 00034           CLRQ    REVISION_XAB                                 ; 0983
                       00000000G   00   D4 00037 1$:       CLRL    FDL$GL_STMNTNUM                              ; 0990
                       00000000G   00   D4 0003D           CLRL    FDL$AB_PRICTRL                               ; 0991
```

```
                                                                      K  5
FDLPARSE         VAX-11 FDL Utilities                               16-Sep-1984 01:50:08    VAX-11 Bliss-32 V4.0-742                Page  9
V04-000          INIT_PARSE                                         14-Sep-1984 12:31:19    DISK$VMSMASTER:[FDL.SRC]FDLPARSE.B32;1   (5)

                                            F8   A6  D4 00043        CLRL    CURRENT_XAB                                        ; 0992
                                            F4   A6  94 00046        CLRB    HIGHEST_AREA_NO                                    ; 0993
                                 6E       0100   8F  3C 00049        MOVZWL  #256, BYTES                                        ; 0998
                                                 58  DD 0004E        PUSHL   R8                                                 ; 1000
                                            04   AE  9F 00050        PUSHAB  BYTES                                              ;
                          00000000G  00          02  FB 00053        CALLS   #2, LIB$GET_VM                                     ;
                                     0D          50  E8 0005A        BLBS    R0, 2$                                             ;
                                     00000000G   8F  DD 0005D        PUSHL   #FDL$_INSVIRMEM                                    ; 1002
                          00000000G  00          01  FB 00063        CALLS   #1, LIB$STOP                                       ;
                                     50          68  D0 0006A 2$:    MOVL    FDL$AB_AREA_BKZ, R0                                ; 1006
          6E             00          6E          00  2C 0006D        MOVC5   #0, (SP), #0, BYTES, (R0)                          ;
                                                 60     00072                                                                   ;
                                                     04 00073        RET                                                        ; 1010

; Routine Size:  116 bytes,    Routine Base:  _FDL$CODE + 0000
```

```
1 %SBTTL  'FINISH_PARSE'
1 GLOBAL ROUTINE  FDL$$FINISH_PARSE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Ties up any loose ends and returns with the final status value
1 !
1 ! Calling Sequence:
1 !
1 !       status = fdl$$finish_parse()
1 !
1 ! Input Parameters:
1 !
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       none
1 !
1 ! Output Parameters:
1 !
1 !       none
1 !
1 ! Implicit Outputs:
1 !
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       SS$_NORMAL      - If everything completed corectly
1 !       FDL$_WARNING    - If there were warnings duing processing
1 !       FDL$_FDLERROR   - If there were real problems
1 !
1 ! Routines Called:
1 !
1 !       lib$free_vm
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         STATUS,
2         XAB     : REF BLOCK [ ,BYTE ],
2         BYTES;
2
2     ! If successful then continue and return ok
2     !
2     IF .FDL$AB_CTRL [ FDL$V_STATUS ]
2     THEN
2         STATUS = SS$_NORMAL
2     ELSE
2
```

```
2             ! If the problem was a warning then continue and return fdl$_warning
2             ! else return imeditaly
2             !
2            IF .FDL$AB_CTRL [ FDL$V_STATUS ] EQLU STS$K_WARNING
2            THEN
2                STATUS = FDL$_WARNING
2            ELSE
2                RETURN FDL$_FDLERROR;
2
2        ! Travel through the xabs and fix up random things
2        ! UNLESS THIS IS JUST A DEFAULT PARSE
2        !
3        IF (
4        ( NOT .FDL$AB_CTRL [ FDL$V_DFLT_PRES ] )
3        OR
4        ( .FDL$AB_CTRL [ FDL$V_REPARSE ] )
2        ) THEN
3            BEGIN
3
3            XAB = .FDL$AB_PARSED_FAB [ FAB$L_XAB ];
3
3            WHILE .XAB NEQU 0
3            DO
4                BEGIN
4
4                ! If this is a key xab fix the fill factors if neccary
4                !
4                IF .XAB [ XAB$B_COD ] EQLU XAB$C_KEY
4                THEN
5                    BEGIN
5
5                    ! Make sure the area numbers are valid if not simply exit
5                    ! RMS will catch it during the create
5                    !
5                    IF ( .XAB [ XAB$B_DAN ] GTRU .HIGHEST_AREA_NO ) OR
6                        ( .XAB [ XAB$B_IAN ] GTRU .HIGHEST_AREA_NO )
5                    THEN
5                        EXITLOOP;
5
5                    ! Data level fill
5                    !
6                    XAB [ XAB$W_DFL ] = ( .FDL$AB_AREA_BKZ [ .XAB [ XAB$B_DAN ] ] * BLOCK_SIZE *
5                                                          .XAB [ XAB$W_DFL ] ) / 100;
5
5                    ! Index level fill
5                    !
6                    XAB [ XAB$W_IFL ] = ( .FDL$AB_AREA_BKZ [ .XAB [ XAB$B_IAN ] ] * BLOCK_SIZE *
5                                                          .XAB [ XAB$W_IFL ] ) / 100
4                    END;
4
4                XAB = .XAB [ XAB$L_NXT ]
4
3                END;
3
2            END;
2
2        ! Deallocate memory for the area bucket size array
```

```
;  408      1125  2     !
;  409      1126  2     BYTES = 256;
;  410      1127  3         BEGIN
;  411      1128  3         LOCAL STATUS;
;  412      1129  3
;  413      1130  4         IF NOT ( STATUS = LIB$FREE_VM ( BYTES,FDL$AB_AREA_BKZ ))
;  414      1131  3         THEN
;  415      1132  3             SIGNAL_STOP ( .STATUS );
;  416      1133  2         END;
;  417      1134  2
;  418      1135  2     RETURN .STATUS
;  419      1136  2
;  420      1137  1     END;



                                      007C 00000        .ENTRY  FDL$$FINISH_PARSE, Save R2,R3,R4,R5,R6          ; 1012
                  56 00000000G  00    9E 00002        MOVAB   FDL$AB_AREA_BKZ, R6                             ;
                  55 00000000G  00    9E 00009        MOVAB   FDL$AB_CTRL, R5                                 ;
                  5E            04    C2 00010        SUBL2   #4, SP                                          ;
   50        65   03            00    EF 00013        EXTZV   #0, #3, FDL$AB_CTRL, R0                         ; 1063
                  05            50    E9 00018        BLBC    R0, 1$                                          ;
                  53            01    D0 0001B        MOVL    #1, STATUS                                      ; 1065
                                13    11 0001E        BRB     3$                                              ;
                                09    12 00020 1$:    BNEQ    2$                                              ; 1071
                  53 00000000G  8F    D0 00022        MOVL    #FDL$_WARNING, STATUS                           ; 1073
                                08    11 00029        BRB     3$                                              ;
                  50 00000000G  8F    D0 0002B 2$:    MOVL    #FDL$_FDLERROR, R0                              ; 1075
                                      04 00032        RET                                                     ;
       04    02   A5            01    E1 00033 3$:    BBC     #1, FDL$AB_CTRL+2, 4$                           ; 1081
                  6B        02  A5    E9 00038        BLBC    FDL$AB_CTRL+2, 7$                               ; 1083
                  50 00000000G  00    D0 0003C 4$:    MOVL    FDL$AB_PARSED_FAB, R0                           ; 1087
                  50        24  A0    D0 00043        MOVL    36(R0), XAB                                     ;
                                5E    13 00047 5$:    BEQL    7$                                              ; 1089
                  15            60    91 00049        CMPB    (XAB), #21                                      ; 1095
                                53    12 0004C        BNEQ    6$                                              ;
                  52        0A  A0    9A 0004E        MOVZBL  10(XAB), R2                                     ; 1102
                  51 00000000'  00    9A 00052        MOVZBL  HIGHEST_AREA_NO, R1                             ;
                  51            52    D1 00059        CMPL    R2, R1                                          ;
                                49    1A 0005C        BGTRU   7$                                              ;
                  51        08  A0    91 0005E        CMPB    8(XAB), R1                                      ; 1103
                                43    1A 00062        BGTRU   7$                                              ;
                  51            66    D0 00064        MOVL    FDL$AB_AREA_BKZ, R1                             ; 1109
                  52      6241        9A 00067        MOVZBL  (R2)[R1], R2                                    ; 1110
                  54        1C  A0    3C 0006B        MOVZWL  28(XAB), R4                                     ;
                  52            54    C4 0006F        MULL2   R4, R2                                          ;
       52         52            09    78 00072        ASHL    #9, R2, R2                                      ; 1109
       54         52 00000064   8F    C7 00076        DIVL3   #100, R2, R4                                    ; 1110
             1C   A0            54    B0 0007E        MOVW    R4, 28(XAB)                                     ;
                  52        08  A0    9A 00082        MOVZBL  8(XAB), R2                                      ; 1114
                  51      6241        9A 00086        MOVZBL  (R2)[R1], R1                                    ; 1115
                  54        1A  A0    3C 0008A        MOVZWL  26(XAB), R4                                     ;
                  51            54    C4 0008E        MULL2   R4, R1                                          ;
       51         51            09    78 00091        ASHL    #9, R1, R1                                      ; 1114
       52         51 00000064   8F    C7 00095        DIVL3   #100, R1, R2                                    ; 1115
```

```
                 1A   A0            52  B0 0009D           MOVW    R2, 26(XAB)                        ;
                      50       04   A0  D0 000A1  6$:      MOVL    4(XAB), XAB                        ; 1118
                                    A0  11 000A5           BRB     5$                                 ;
                      6E     0100   8F  3C 000A7  7$:      MOVZWL  #256, BYTES                        ; 1126
                                    56  DD 000AC           PUSHL   R6                                 ; 1130
                               04   AE  9F 000AE           PUSHAB  BYTES                              ;
          00000000G   00            02  FB 000B1           CALLS   #2, LIB$FREE_VM                    ;
                      09            50  E8 000B8           BLBS    STATUS, 8$                         ;
                                    50  DD 000BB           PUSHL   STATUS                             ; 1132
          00000000G   00            01  FB 000BD           CALLS   #1, LIB$STOP                       ;
                      50            53  D0 000C4  8$:      MOVL    STATUS, R0                         ; 1135
                                        04 000C7           RET                                        ; 1137

; Routine Size:  200 bytes,    Routine Base:  _FDL$CODE + 0074
```

```
%SBTTL  'LINE_PARSED'
GLOBAL ROUTINE  FDL$$LINE_PARSED =
!++
!
! Functional Description:
!
!       Main parsing routine.  Called by the parse tables it in turn
!       calles the appropiate routines to parse the fdl line.
!
! Calling Sequence:
!
!       Called from parse tables
!
! Input Parameters:
!
!       fdl$gl_primary  - Primary code
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!
!       Values returned by called routines
!
! Routines Called:
!
!       .fdl$gl_pcall
!       set_area_p
!       set_date_p
!       set_jnl_p
!       set_acl_p        not supported V4.0
!       set_file_p
!       set_key_p
!       set_record_p
!       set_access_p
!       set_sharing_p
!       set_connect_p
!
! Side Effects:
!       none
!
!--

    BEGIN

    TPARSE_ARGS;

    LOCAL
        STATUS;

    STATUS = SS$_NORMAL;
    ! If we have processed some really bad stuff then dont bother
    !
    IF .FDL$AB_CTRL [ FDL$V_STATUS ] EQLU STS$K_ERROR
    THEN
        RETURN .STATUS;

    ! If this is an EDF call then let them process the command
    !
    IF .FDL$AB_CTRL [ FDL$V_PCALL ]
    THEN
        STATUS = (.FDL$GL_PCALL)()
    ELSE

        ! If this is a primary only or line comment call ignore it
        !
        IF NOT ( .FDL$AB_CTRL [ FDL$V_NEWPRI ] OR .FDL$AB_CTRL [ FDL$V_LINECMT ] )
        THEN
            CASE .FDL$GL_PRIMARY FROM FDL$C_ACCESS TO FDL$C_TITLE OF
                          SET

                          [ FDL$C_ACCESS ] : SET_ACCESS_P();

                          [ FDL$C_ACL ]    : SET_ACL_P();

                          [ FDL$C_AREA ]   : SET_AREA_P();

                          [ FDL$C_CONNECT ] : SET_CONNECT_P();

                          [ FDL$C_DATE ]   : SET_DATE_P();

                          [ FDL$C_FILE ]   : SET_FILE_P();

                          [ FDL$C_JNL ]    : SET_JNL_P();

                          [ FDL$C_KEY ]    : SET_KEY_P();

                          [ FDL$C_RECORD ] : SET_RECORD_P();

                          [ FDL$C_SHARING ] : SET_SHARING_P();

                          [ INRANGE ]      : 0;    ! Catch all for non usefull
                                                   ! primaries

                          TES;

    ! Clear new primary in case it was set
    !
    FDL$AB_CTRL [ FDL$V_NEWPRI ] = _CLEAR;

    RETURN .STATUS

    END;
```

```
                                             000C 00000         .ENTRY  FDL$$LINE_PARSED, Save R2,R3                 ; 1139
                          53 00000000G   00    9E 00002         MOVAB   FDL$AB_CTRL, R3                              ;
                          52               01  D0 00009         MOVL    #1, STATUS                                   ; 1194
      02          63      03               00  ED 0000C         CMPZV   #0, #3, FDL$AB_CTRL, #2                      ; 1198
                                           03  12 00011         BNEQ    1$                                           ;
                                         009E  31 00013         BRW     16$                                          ;
                  0F         01   A3       02  E1 00016 1$:     BBC     #2, FDL$AB_CTRL+1, 2$                        ; 1204
                          50 00000000G     00  D0 0001B         MOVL    FDL$GL_PCALL, R0                             ; 1206
                          60               00  FB 00022         CALLS   #0, (R0)                                     ;
                          52               50  D0 00025         MOVL    R0, STATUS                                   ;
                                           7E  11 00028         BRB     13$                                          ;
                  7A      63               05  E0 0002A 2$:     BBS     #5, FDL$AB_CTRL, 13$                         ; 1211
                  7E         01   A3       01  E0 0002E         BBS     #1, FDL$AB_CTRL+1, 15$                       ;
                  0E      01 00000000G     00  CF 00033         CASEL   FDL$GL_PRIMARY, #1, #14                      ; 1213
    0076        0076    0027             001E     0003B 3$:     .WORD   4$-3$,-                                      ;
    004B        0042    0039             0030     00043                 5$-3$,-                                      ;
    0066        005D    0054             0076     0004B                 15$-3$,-                                     ;
                0076    0076             006F     00053                 15$-3$,-                                     ;
                                                                        6$-3$,-                                      ;
                                                                        7$-3$,-                                      ;
                                                                        8$-3$,-                                      ;
                                                                        9$-3$,-                                      ;
                                                                        15$-3$,-                                     ;
                                                                        10$-3$,-                                     ;
                                                                        11$-3$,-                                     ;
                                                                        12$-3$,-                                     ;
                                                                        14$-3$,-                                     ;
                                                                        15$-3$,-                                     ;
                                                                        15$-3$                                       ;
                   00000000V  00           00  FB 00059 4$:     CALLS   #0, SET_ACCESS_P                             ; 1216
                                           4F  11 00060         BRB     15$                                          ;
                   00000000V  00           00  FB 00062 5$:     CALLS   #0, SET_ACL_P                                ; 1218
                                           46  11 00069         BRB     15$                                          ;
                   00000000V  00           00  FB 0006B 6$:     CALLS   #0, SET_AREA_P                               ; 1220
                                           3D  11 00072         BRB     15$                                          ;
                   00000000V  00           00  FB 00074 7$:     CALLS   #0, SET_CONNECT_P                            ; 1222
                                           34  11 0007B         BRB     15$                                          ;
                   00000000V  00           00  FB 0007D 8$:     CALLS   #0, SET_DATE_P                               ; 1224
                                           2B  11 00084         BRB     15$                                          ;
                   00000000V  00           00  FB 00086 9$:     CALLS   #0, SET_FILE_P                               ; 1226
                                           22  11 0008D         BRB     15$                                          ;
                   00000000V  00           00  FB 0008F 10$:    CALLS   #0, SET_JNL_P                                ; 1228
                                           19  11 00096         BRB     15$                                          ;
                   00000000V  00           00  FB 00098 11$:    CALLS   #0, SET_KEY_P                                ; 1230
                                           10  11 0009F         BRB     15$                                          ;
                   00000000V  00           00  FB 000A1 12$:    CALLS   #0, SET_RECORD_P                             ; 1232
                                           07  11 000A8 13$:    BRB     15$                                          ;
                   00000000V  00           00  FB 000AA 14$:    CALLS   #0, SET_SHARING_P                            ; 1234
                              63           20  8A 000B1 15$:    BICB2   #32, FDL$AB_CTRL                             ; 1243
                              50           52  D0 000B4 16$:    MOVL    STATUS, R0                                   ; 1245
                                           04     000B7         RET                                                  ; 1247

; Routine Size:  184 bytes,    Routine Base:  _FDL$CODE + 013C
```

```
%SBTTL 'SET_AREA_P'
ROUTINE SET_AREA_P : NOVALUE =
!++
!
! Functional Description:
!
!       Fill in the blanks for the allocation xab
!
! Calling Sequence:
!
!       set_area_p()
!
! Input Parameters:
!       none
!
! Implicit Inputs:
!
!       fdl$secondary    - Secondary code
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!       none
!
! Routines Called:
!
!       allocate_xab
!
! Side Effects:
!       none
!
!--

    BEGIN

    ! To aviod some duplication of code ....
    ! Find out if there is a current xab  if not then get one
    ! OR If the current xab is not the same type or number of what we want
    ! then get a new one
    !
    IF ( IF .CURRENT_XAB EQLU 0
         THEN 1
         ELSE
         IF ( .CURRENT_XAB [ XAB$B_COD ] NEQ XAB$C_ALL ) OR
            ( .CURRENT_XAB [ XAB$B_AID ] NEQ .FDL$GL_PRINUM )
         THEN 1
         ELSE 0 )
    THEN
        BEGIN

        ! Allocate memory for the new xab
        !
        ALLOCATE_XAB ( XAB$C_ALL, .FDL$GL_PRINUM );
```

```

                ! Set the area number in the xab
                !
                CURRENT_XAB [ XAB$B_AID ] = .FDL$GL_PRINUM;

                ! If this is area 0 then copy the allocation etc. from the fab (this
                ! is because using areas overide the fab allocation and this
                ! makes it look like it doesen't)
                !
                IF .CURRENT_XAB [ XAB$B_AID ] EQLU 0
                THEN
                    BEGIN

                    ! Copy Allocation, Bucket size and Extention
                    !
                    CURRENT_XAB [ XAB$L_ALQ ] = .FDL$AB_PARSED_FAB [ FAB$L_ALQ ];
                    CURRENT_XAB [ XAB$B_BKZ ] = .FDL$AB_PARSED_FAB [ FAB$B_BKS ];
                    CURRENT_XAB [ XAB$W_DEQ ] = .FDL$AB_PARSED_FAB [ FAB$W_DEQ ];
                    CURRENT_XAB [ XAB$L_ALQ ] = .FDL$AB_PARSED_FAB [ FAB$L_ALQ ];

                    IF .FDL$AB_PARSED_FAB [ FAB$B_BKS ] NEQU 0
                    THEN
                        FDL$AB_AREA_BKZ [ 0 ] = .FDL$AB_PARSED_FAB [ FAB$B_BKS ]
                    ELSE
                        FDL$AB_AREA_BKZ [ 0 ] = BUCKET_DEFAULT;

                    ! Also get the duplicated contigous options:
                    !
                    ! Contigous best try
                    !
                    IF .FDL$AB_PARSED_FAB [ FAB$V_CBT ]
                    THEN
                        CURRENT_XAB [ XAB$V_CBT ] = _SET;

                    ! Contigous
                    !
                    IF .FDL$AB_PARSED_FAB [ FAB$V_CTG ]
                    THEN
                        CURRENT_XAB [ XAB$V_CTG ] = _SET

                    END
                ELSE

                    ! Count this area
                    !
                    HIGHEST_AREA_NO = .HIGHEST_AREA_NO + 1

                END;

            ! Set the fields in the area xab
            !
            CASE .FDL$GL_SECONDARY FROM FDL$C_ALLOC TO FDL$C_VOLU OF
            SET
                [ FDL$C_ALLOC ] : CURRENT_XAB [ XAB$L_ALQ ] = .FDL$GL_NUMBER;

                [ FDL$C_BTCONT ]: CURRENT_XAB [ XAB$V_CBT ] = .FDL$GL_SWITCH;

```

```
[ FDL$C_BKT ]   : BEGIN

                  CURRENT_XAB [ XAB$B_BKZ ] = .FDL$GL_NUMBER;

                  ! Fill in the table for figuring fill numbers latter
                  !
                  FDL$AB_AREA_BKZ [ .FDL$GL_PRINUM ] = .FDL$GL_NUMBER

                  END;

[ FDL$C_CONTG ] : CURRENT_XAB [ XAB$V_CTG ] = .FDL$GL_SWITCH;

[ FDL$C_EXACT ] : CURRENT_XAB [ XAB$V_HRD ] = .FDL$GL_SWITCH;

[ FDL$C_EXTND ] : CURRENT_XAB [ XAB$W_DEQ ] = .FDL$GL_NUMBER;

[ FDL$C_POSI ]  : CASE .FDL$GL_QUALIFIER FROM
                                        FDL$C_ANYPOS TO FDL$C_VIRPOS OF
          SET
              [ FDL$C_ANYPOS ] : CURRENT_XAB [ XAB$V_ONC ] = _SET;

              [ FDL$C_CLUSPOS ] : CURRENT_XAB [ XAB$V_ONC ] = _SET;

              [ FDL$C_CYLPOS ] : BEGIN
                                 CURRENT_XAB [ XAB$B_ALN ] = XAB$C_CYL;
                                 CURRENT_XAB [ XAB$L_LOC ] = .FDL$GL_NUMBER
                                 END;

              [ FDL$C_FIDPOS ] : BEGIN
                                 CURRENT_XAB [ XAB$W_RFI0 ] = .FDL$GL_FID1;
                                 CURRENT_XAB [ XAB$W_RFI2 ] = .FDL$GL_FID2;
                                 CURRENT_XAB [ XAB$W_RFI4 ] = .FDL$GL_FID3
                                 END;

              [ FDL$C_FNMPOS ] : BEGIN
                                 FIND_ID();
                                 CURRENT_XAB [ XAB$W_RFI0 ] = .FDL$GL_FID1;
                                 CURRENT_XAB [ XAB$W_RFI2 ] = .FDL$GL_FID2;
                                 CURRENT_XAB [ XAB$W_RFI4 ] = .FDL$GL_FID3
                                 END;

              [ FDL$C_LOGPOS ] : BEGIN
                                 CURRENT_XAB [ XAB$B_ALN ] = XAB$C_LBN;
                                 CURRENT_XAB [ XAB$L_LOC ] = .FDL$GL_NUMBER
                                 END;

              [ FDL$C_NOPOS ]  : CURRENT_XAB [ XAB$B_ALN ] = _CLEAR;

              [ FDL$C_VIRPOS ] : BEGIN
                                 CURRENT_XAB [ XAB$B_ALN ] = XAB$C_VBN;
                                 CURRENT_XAB [ XAB$L_LOC ] = .FDL$GL_NUMBER
                                 END;
          TES;

[ FDL$C_VOLU ]  : BEGIN
                  CURRENT_XAB [ XAB$W_VOL ] = .FDL$GL_NUMBER;
```

```
;  704      1419  3                          ! If the guy didn't give any placement do it for him
;  705      1420  3                          !
;  706      1421  3                          IF .CURRENT_XAB [ XAB$B_ALN ] EQLU _CLEAR
;  707      1422  3                          THEN
;  708      1423  3                              CURRENT_XAB [ XAB$B_ALN ] = XAB$C_LBN;
;  709      1424  3
;  710      1425  2                          END;
;  711      1426  2
;  712      1427  2         TES;
;  713      1428  2
;  714      1429  2         RETURN
;  715      1430  2
;  716      1431  1         END;



                                          00FC 00000 SET_AREA_P:
                                                          .WORD    Save R2,R3,R4,R5,R6,R7                      ; 1249
                         57 00000000G  00   9E 00002      MOVAB    FDL$AB_AREA_BKZ, R7                         ;
                         56 00000000G  00   9E 00009      MOVAB    FDL$GL_SWITCH, R6                           ;
                         55 00000000G  00   9E 00010      MOVAB    FDL$GL_PRINUM, R5                           ;
                         54 00000000G  00   9E 00017      MOVAB    FDL$GL_NUMBER, R4                           ;
                         53 00000000'  00   9E 0001E      MOVAB    CURRENT_XAB, R3                             ;
                         50            63   D0 00025      MOVL     CURRENT_XAB, R0                             ; 1292
                                       0D   13 00028      BEQL     1$                                          ;
                         14            60   91 0002A      CMPB     (R0), #20                                   ; 1295
                                       08   12 0002D      BNEQ     1$                                          ;
   65      17    A0      08            00   ED 0002F      CMPZV    #0, #8, 23(R0), FDL$GL_PRINUM               ; 1296
                                       58   13 00035      BEQL     6$                                          ;
                                       65   DD 00037 1$:  PUSHL    FDL$GL_PRINUM                               ; 1304
                                       14   DD 00039      PUSHL    #20                                         ;
                00000000V  00          02   FB 0003B      CALLS    #2, ALLOCATE_XAB                            ;
                           51          63   D0 00042      MOVL     CURRENT_XAB, R1                             ; 1308
                      17   A1          65   90 00045      MOVB     FDL$GL_PRINUM, 23(R1)                       ;
                                       41   12 00049      BNEQ     5$                                          ; 1314
                         50 00000000G  00   D0 0004B      MOVL     FDL$AB_PARSED_FAB, R0                       ; 1320
                      10   A1       10 A0   D0 00052      MOVL     16(R0), 16(R1)                              ;
                      16   A1       3E A0   90 00057      MOVB     62(R0), 22(R1)                              ; 1321
                      14   A1       14 A0   B0 0005C      MOVW     20(R0), 20(R1)                              ; 1322
                      10   A1       10 A0   D0 00061      MOVL     16(R0), 16(R1)                              ; 1323
                           52          67   D0 00066      MOVL     FDL$AB_AREA_BKZ, R2                         ; 1327
                                    3E A0   95 00069      TSTB     62(R0)                                      ; 1325
                                       06   13 0006C      BEQL     2$                                          ;
                           62       3E A0   90 0006E      MOVB     62(R0), (R2)                                ; 1327
                                       03   11 00072      BRB      3$                                          ;
                           62          02   90 00074 2$:  MOVB     #2, (R2)                                    ; 1329
                 04   06   A0          05   E1 00077 3$:  BBC      #5, 6(R0), 4$                               ; 1335
                      08   A1          20   88 0007C      BISB2    #32, 8(R1)                                  ; 1337
                 0A   06   A0          04   E1 00080 4$:  BBC      #4, 6(R0), 6$                               ; 1341
                      08   A1       80 8F   88 00085      BISB2    #128, 8(R1)                                 ; 1343
                                       03   11 0008A      BRB      6$                                          ; 1341
                                    FC A3   96 0008C 5$:  INCB     HIGHEST_AREA_NO                             ; 1350
                 07         1B 00000000G  00   CF 0008F 6$:  CASEL    FDL$GL_SECONDARY, #27, #7                ; 1356
   0034          0022       0018       0010    00097 7$:  .WORD    8$-7$,-                                     ;
   00B8          0050       0048       003E    0009F               9$-7$,-                                     ;
```

```
                                                                     10$-7$,-                                          ;
                                                                     11$-7$,-                                          ;
                                                                     12$-7$,-                                          ;
                                                                     13$-7$,-                                          ;
                                                                     14$-7$,-                                          ;
                                                                     24$-7$                                            ;
                         50               63  D0 000A7 8$:   MOVL    CURRENT_XAB, R0                                   ; 1358
                   10    A0               64  D0 000AA       MOVL    FDL$GL_NUMBER, 16(R0)                             ;
                                          04 000AE           RET                                                       ;
                         50               63  D0 000AF 9$:   MOVL    CURRENT_XAB, R0                                   ; 1360
  08   A0       01       05               66  F0 000B2       INSV    FDL$GL_SWITCH, #5, #1, 8(R0)                      ;
                                          04 000B8           RET                                                       ;
                         50               63  D0 000B9 10$:  MOVL    CURRENT_XAB, R0                                   ; 1364
                         51               64  D0 000BC       MOVL    FDL$GL_NUMBER, R1                                 ;
                   16    A0               51  90 000BF       MOVB    R1, 22(R0)                                        ;
                50       67               65  C1 000C3       ADDL3   FDL$GL_PRINUM, FDL$AB_AREA_BKZ, R0                ; 1368
                         60               51  90 000C7       MOVB    R1, (R0)                                          ;
                                          04 000CA           RET                                                       ;
                         50               63  D0 000CB 11$:  MOVL    CURRENT_XAB, R0                                   ; 1372
  08   A0       01       07               66  F0 000CE       INSV    FDL$GL_SWITCH, #7, #1, 8(R0)                      ;
                                          04 000D4           RET                                                       ;
                         50               63  D0 000D5 12$:  MOVL    CURRENT_XAB, R0                                   ; 1374
  08   A0       01       00               66  F0 000D8       INSV    FDL$GL_SWITCH, #0, #1, 8(R0)                      ;
                                          04 000DE           RET                                                       ;
                         50               63  D0 000DF 13$:  MOVL    CURRENT_XAB, R0                                   ; 1376
                   14    A0               64  B0 000E2       MOVW    FDL$GL_NUMBER, 20(R0)                             ;
                                          04 000E6           RET                                                       ;
                07       00 00000000G     00  CF 000E7 14$:  CASEL   FDL$GL_QUALIFIER, #0, #7                          ; 1378
   0028       0018     0010             0010     000EF 15$:  .WORD   16$-15$,-                                         ;
   0054       004D     0044             0021     000F7               16$-15$,-                                         ;
                                                                     17$-15$,-                                         ;
                                                                     19$-15$,-                                         ;
                                                                     18$-15$,-                                         ;
                                                                     20$-15$,-                                         ;
                                                                     21$-15$,-                                         ;
                                                                     22$-15$                                           ;
                         50               63  D0 000FF 16$:  MOVL    CURRENT_XAB, R0                                   ; 1383
                   08    A0               02  88 00102       BISB2   #2, 8(R0)                                         ;
                                          04 00106           RET                                                       ;
                         50               63  D0 00107 17$:  MOVL    CURRENT_XAB, R0                                   ; 1386
                   09    A0               01  90 0010A       MOVB    #1, 9(R0)                                         ;
                                          3A  11 0010E       BRB     23$                                               ; 1387
             00000000V   00               00  FB 00110 18$:  CALLS   #0, FIND_ID                                       ; 1397
                         50               63  D0 00117 19$:  MOVL    CURRENT_XAB, R0                                   ; 1398
                   18    A0 00000000G     00  B0 0011A       MOVW    FDL$GL_FID1, 24(R0)                               ;
                   1A    A0 00000000G     00  B0 00122       MOVW    FDL$GL_FID2, 26(R0)                               ; 1399
                   1C    A0 00000000G     00  B0 0012A       MOVW    FDL$GL_FID3, 28(R0)                               ; 1400
                                          04 00132           RET                                                       ;
                         50               63  D0 00133 20$:  MOVL    CURRENT_XAB, R0                                   ; 1404
                   09    A0               02  90 00136       MOVB    #2, 9(R0)                                         ;
                                          0E  11 0013A       BRB     23$                                               ; 1405
                         50               63  D0 0013C 21$:  MOVL    CURRENT_XAB, R0                                   ; 1408
                                    09    A0  94 0013F       CLRB    9(R0)                                             ;
                                          04 00142           RET                                                       ;
                         50               63  D0 00143 22$:  MOVL    CURRENT_XAB, R0                                   ; 1411
                   09    A0               03  90 00146       MOVB    #3, 9(R0)                                         ;
                   0C    A0               64  D0 0014A 23$:  MOVL    FDL$GL_NUMBER, 12(R0)                             ; 1412
```

```
                                   04 0014E        RET                                      ; 1378
              50                63 D0 0014F 24$:   MOVL    CURRENT_XAB, R0                  ; 1417
        0A    A0                64 B0 00152        MOVW    FDL$GL_NUMBER, 10(R0)            ;
                          09    A0 95 00156        TSTB    9(R0)                            ; 1421
                                04 12 00159        BNEQ    25$                              ;
        09    A0                02 90 0015B        MOVB    #2, 9(R0)                        ; 1423
                                   04 0015F 25$:   RET                                      ; 1431

; Routine Size:  352 bytes,     Routine Base:  _FDL$CODE + 01F4
```

```
1 %SBTTL 'SET_DATE_P'
1 ROUTINE SET_DATE_P : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Fill in the blanks for the revision date and time xab
1 !
1 ! Calling Sequence:
1 !
1 !       set_date_p()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       fdl$secondary   - Secondary code
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routines Called:
1 !
1 !       sys$bintim
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     ! See which xab we need
2     !
2     IF .FDL$GL_SECONDARY EQLU FDL$C_REV
2     THEN
3         BEGIN
3
3         ! If the revision xab has not been connected then connect it
3         !
3         IF .REVISION_XAB EQLU 0
3         THEN
3
3             ! Allocate the xab an enter it into the chain
3             !
3             REVISION_XAB = ALLOCATE_XAB ( XAB$C_RDT, 0 )
3
3         END
2     ELSE
2
```

```
;  775      1489  2          ! If the date xab has not been allocated then get one
;  776      1490  2          !
;  777      1491  2          IF .DATE_XAB EQLU 0
;  778      1492  2          THEN
;  779      1493  2
;  780      1494  2              ! Allocate the xab an enter it into the chain
;  781      1495  2              !
;  782      1496  2              DATE_XAB = ALLOCATE_XAB ( XAB$C_DAT, 0 );
;  783      1497  2
;  784      1498  2      ! Fill in the correct field
;  785      1499  2      !
;  786      1500  2      CASE .FDL$GL_SECONDARY FROM FDL$C_BACKUP TO FDL$C_REV OF
;  787      1501  2      SET
;  788      1502  3          [ FDL$C_BACKUP ]: BEGIN
;  789      1503  3                          DATE_XAB [ XAB$L_BDT0 ] = .FDL$AL_DATE_TIME [ 0 ];
;  790      1504  3                          DATE_XAB [ XAB$L_BDT4 ] = .FDL$AL_DATE_TIME [ 1 ]
;  791      1505  2                          END;
;  792      1506  2
;  793      1507  3          [ FDL$C_CREAT ] : BEGIN
;  794      1508  3                          DATE_XAB [ XAB$L_CDT0 ] = .FDL$AL_DATE_TIME [ 0 ];
;  795      1509  3                          DATE_XAB [ XAB$L_CDT4 ] = .FDL$AL_DATE_TIME [ 1 ]
;  796      1510  2                          END;
;  797      1511  2
;  798      1512  3          [ FDL$C_EXPR ]  : BEGIN
;  799      1513  3                          DATE_XAB [ XAB$L_EDT0 ] = .FDL$AL_DATE_TIME [ 0 ];
;  800      1514  3                          DATE_XAB [ XAB$L_EDT4 ] = .FDL$AL_DATE_TIME [ 1 ]
;  801      1515  2                          END;
;  802      1516  2
;  803      1517  3          [ FDL$C_REV ]   : BEGIN
;  804      1518  3                          REVISION_XAB [ XAB$L_RDT0 ] = .FDL$AL_DATE_TIME [ 0 ];
;  805      1519  3                          REVISION_XAB [ XAB$L_RDT4 ] = .FDL$AL_DATE_TIME [ 1 ]
;  806      1520  2                          END;
;  807      1521  2
;  808      1522  2      TES;
;  809      1523  2
;  810      1524  2      RETURN
;  811      1525  2
;  812      1526  1      END;




                                           003C 00000 SET_DATE_P:
                                                                  .WORD   Save R2,R3,R4,R5                       ; 1433
                     55 00000000G  00  9E 00002                   MOVAB   FDL$GL_SECONDARY, R5                   ;
                     54 00000000V  00  9E 00009                   MOVAB   ALLOCATE_XAB, R4                       ;
                     53 00000000'  00  9E 00010                   MOVAB   DATE_XAB, R3                           ;
        00000047     8F            65  D1 00017                   CMPL    FDL$GL_SECONDARY, #71                  ; 1473
                                   11  12 0001E                   BNEQ    1$                                     ;
                          04  A3   D5 00020                       TSTL    REVISION_XAB                           ; 1479
                                   19  12 00023                   BNEQ    2$                                     ;
                     7E            1E  7D 00025                   MOVQ    #30, -(SP)                             ; 1484
                     64            02  FB 00028                   CALLS   #2, ALLOCATE_XAB                       ;
             04      A3            50  D0 0002B                   MOVL    R0, REVISION_XAB                       ;
                                   0D  11 0002F                   BRB     2$                                     ; 1475
                                   63  D5 00031 1$:               TSTL    DATE_XAB                               ; 1491
```

```
                                              09  12 00033          BNEQ    2$                                     ;
                           7E                 12  7D 00035          MOVQ    #18, -(SP)                             ; 1496
                           64                 02  FB 00038          CALLS   #2, ALLOCATE_XAB                       ;
                           63                 50  D0 0003B          MOVL    R0, DATE_XAB                           ;
                           52 00000000G       00  D0 0003E 2$:      MOVL    FDL$AL_DATE_TIME, R2                   ; 1503
                           51 00000000G       00  D0 00045          MOVL    FDL$AL_DATE_TIME+4, R1                 ; 1504
              03 00000044  8F                 65  CF 0004C          CASEL   FDL$GL_SECONDARY, #68, #3              ; 1500
     002C          0020           0014      0008     00054 3$:      .WORD   4$-3$,-                                ;
                                                                            5$-3$,-                                ;
                                                                            6$-3$,-                                ;
                                                                            7$-3$                                  ;
                           50                 63  D0 0005C 4$:      MOVL    DATE_XAB, R0                           ; 1503
                     24    A0                 52  D0 0005F          MOVL    R2, 36(R0)                             ;
                     28    A0                 51  D0 00063          MOVL    R1, 40(R0)                             ; 1504
                                                  04 00067          RET                                            ;
                           50                 63  D0 00068 5$:      MOVL    DATE_XAB, R0                           ; 1508
                     14    A0                 52  D0 0006B          MOVL    R2, 20(R0)                             ;
                     18    A0                 51  D0 0006F          MOVL    R1, 24(R0)                             ; 1509
                                                  04 00073          RET                                            ;
                           50                 63  D0 00074 6$:      MOVL    DATE_XAB, R0                           ; 1513
                     1C    A0                 52  D0 00077          MOVL    R2, 28(R0)                             ;
                     20    A0                 51  D0 0007B          MOVL    R1, 32(R0)                             ; 1514
                                                  04 0007F          RET                                            ;
                           50         04      A3  D0 00080 7$:      MOVL    REVISION_XAB, R0                       ; 1518
                     0C    A0                 52  D0 00084          MOVL    R2, 12(R0)                             ;
                     10    A0                 51  D0 00088          MOVL    R1, 16(R0)                             ; 1519
                                                  04 0008C          RET                                            ; 1526

; Routine Size:  141 bytes,    Routine Base:  _FDL$CODE + 0354
```

```
%SBTTL 'SET_JNL_P'
ROUTINE SET_JNL_P : NOVALUE =
!++
!
! Functional Description:
!
!       Fill in the blanks for the journal xab
!
! Calling Sequence:
!
!       set_jnl_p()
!
! Input Parameters:
!       none
!
! Implicit Inputs:
!
!       fdl$secondary   - Secondary code
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!       none
!
! Routines Called:
!
!       none
!
! Side Effects:
!       none
!
!--

    BEGIN

    ! If the xab has not been connected, then connect it
    !
    IF .JNL_XAB EQLU 0
    THEN
        ! Allocate the xab and enter it into the chain
        !
        JNL_XAB = ALLOCATE_XAB ( XAB$C_JNL, 0 );

    ! Fill in the correct field
    !
    CASE .FDL$GL_SECONDARY FROM FDL$C_AFTIM TO FDL$C_RU OF
    SET
        [ FDL$C_AFTIM ]     : JNL_XAB [ XAB$V_AI ] = .FDL$GL_SWITCH;

        [ FDL$C_AFTNAM ]    : BEGIN

                                ! Allocate a buffer for the string and copy to it
                                !
```

```
                    JNL_XAB [ XAB$L_AIA ] =
                       FDL$$GET_VM( .FDL$AB_STRING [ DSC$W_LENGTH ] );

                    CH$MOVE(    .FDL$AB_STRING [ DSC$W_LENGTH ],
                                .FDL$AB_STRING [ DSC$A_POINTER ],
                                .JNL_XAB [ XAB$L_AIA ] );

                    JNL_XAB [ XAB$B_AIS ] =
                                .FDL$AB_STRING [ DSC$W_LENGTH ]
                    END;

[ FDL$C_AUDIT ]   : JNL_XAB [ XAB$V_AT ] = .FDL$GL_SWITCH;

[ FDL$C_AUDNAM ]  : BEGIN
                    ! Allocate a buffer for the string and copy to it
                    !
                    JNL_XAB [ XAB$L_ATA ] =
                       FDL$$GET_VM( .FDL$AB_STRING [ DSC$W_LENGTH ] );

                    CH$MOVE(    .FDL$AB_STRING [ DSC$W_LENGTH ],
                                .FDL$AB_STRING [ DSC$A_POINTER ],
                                .JNL_XAB [ XAB$L_ATA ] );

                    JNL_XAB [ XAB$B_ATS ] =
                                .FDL$AB_STRING [ DSC$W_LENGTH ]
                    END;

[ FDL$C_BEFIM ]   : JNL_XAB [ XAB$V_BI ] = .FDL$GL_SWITCH;

[ FDL$C_BEFNAM ]  : BEGIN
                    ! Allocate a buffer for the string and copy to it
                    !
                    JNL_XAB [ XAB$L_BIA ] =
                       FDL$$GET_VM( .FDL$AB_STRING [ DSC$W_LENGTH ] );

                    CH$MOVE(    .FDL$AB_STRING [ DSC$W_LENGTH ],
                                .FDL$AB_STRING [ DSC$A_POINTER ],
                                .JNL_XAB [ XAB$L_BIA ] );

                    JNL_XAB [ XAB$B_BIS ] =
                                .FDL$AB_STRING [ DSC$W_LENGTH ]
                    END;

[ FDL$C_RU ]      : BEGIN
                    ! Set the recovery unit bit according to what
                    ! was specified
                    !
                    JNL_XAB [ XAB$V_RU ] = _CLEAR;
                    JNL_XAB [ XAB$V_ONLY_RU ] = _CLEAR;
                    JNL_XAB [ XAB$V_NEVER_RU ] = _CLEAR;

                    IF .FDL$GL_QUALIFIER EQLU FDL$C_IF_IN
                    THEN
                        JNL_XAB [ XAB$V_RU ] = _SET

                    ELSE IF .FDL$GL_QUALIFIER EQLU FDL$C_NEC
                    THEN
```

```
FDLPARSE              VAX-11 FDL Utilities                        D  7
V04-000               SET_JNL_P                               16-Sep-1984 01:50:08    VAX-11 Bliss-32 V4.0-742                  Page  28
                                                              14-Sep-1984 12:31:19    DISK$VMSMASTER:[FDL.SRC]FDLPARSE.B32;1    (10)

;  928        1641  3                                         JNL_XAB [ XAB$V_ONLY_RU ] = _SET
;  929        1642  3
;  930        1643  3                                     ELSE IF .FDL$GL_QUALIFIER EQLU FDL$C_NEVER
;  931        1644  3                                     THEN
;  932        1645  3                                         JNL_XAB [ XAB$V_NEVER_RU ] = _SET;
;  933        1646  3
;  934        1647  2                                     END;
;  935        1648  2
;  936        1649  2             TES;
;  937        1650  2
;  938        1651  2             RETURN
;  939        1652  2
;  940        1653  1             END;




                                                             0FFC 00000 SET_JNL_P:
                                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 1528
                                            5B 00000000V   00 9E  00002         MOVAB   FDL$$GET_VM, R11                             ;
                                            5A 00000000G   00 9E  00009         MOVAB   FDL$GL_SWITCH, R10                           ;
                                            59 00000000'   00 9E  00010         MOVAB   JNL_XAB, R9                                  ;
                                            58 00000000G   00 9E  00017         MOVAB   FDL$AB_STRING, R8                            ;
                                                           69 D5  0001E         TSTL    JNL_XAB                                      ; 1568
                                                           0D 12  00020         BNEQ    1$                                           ;
                                            7E             22 7D  00022         MOVQ    #34, -(SP)                                   ; 1572
                                  00000000V 00             02 FB  00025         CALLS   #2, ALLOCATE_XAB                             ;
                                            69             50 D0  0002C         MOVL    R0, JNL_XAB                                  ;
                                            52             69 D0  0002F 1$:     MOVL    JNL_XAB, R2                                  ; 1578
                              06  00000070  8F 00000000G   00 CF  00032         CASEL   FDL$GL_SECONDARY, #112, #6                   ; 1576
              003A          0033          0015           000E     0003E 2$:     .WORD   3$-2$,-                                      ;
                            007D          005F           0058     00046                 4$-2$,-                                      ;
                                                                                        5$-2$,-                                      ;
                                                                                        6$-2$,-                                      ;
                                                                                        7$-2$,-                                      ;
                                                                                        8$-2$,-                                      ;
                                                                                        9$-2$                                        ;
           08   A2            01            03             6A F0  0004C 3$:     INSV    FDL$GL_SWITCH, #3, #1, 8(R2)                 ; 1578
                                                              04  00052         RET                                                  ;
                                            7E             68 3C  00053 4$:     MOVZWL  FDL$AB_STRING, -(SP)                         ; 1585
                                            6B             01 FB  00056         CALLS   #1, FDL$$GET_VM                              ;
                                        18  A2             50 D0  00059         MOVL    R0, 24(R2)                                   ;
                                            57             68 3C  0005D         MOVZWL  FDL$AB_STRING, R7                            ; 1587
                                            50        04   A8 D0  00060         MOVL    FDL$AB_STRING+4, R0                          ; 1588
                                            56             69 D0  00064         MOVL    JNL_XAB, R6                                  ; 1589
                          18  B6            60             57 28  00067         MOVC3   R7, (R0), @24(R6)                            ;
                                        14  A6             57 90  0006C         MOVB    R7, 20(R6)                                   ; 1592
                                                              04  00070         RET                                                  ; 1591
           08   A2            01            04             6A F0  00071 5$:     INSV    FDL$GL_SWITCH, #4, #1, 8(R2)                 ; 1595
                                                              04  00077         RET                                                  ;
                                            7E             68 3C  00078 6$:     MOVZWL  FDL$AB_STRING, -(SP)                         ; 1601
                                            6B             01 FB  0007B         CALLS   #1, FDL$$GET_VM                              ;
                                        20  A2             50 D0  0007E         MOVL    R0, 32(R2)                                   ;
                                            57             68 3C  00082         MOVZWL  FDL$AB_STRING, R7                            ; 1603
                                            50        04   A8 D0  00085         MOVL    FDL$AB_STRING+4, R0                          ; 1604
                                            56             69 D0  00089         MOVL    JNL_XAB, R6                                  ; 1605
           20  B6           60       57  28 0008C          MOVC3   R7, (R0), @32(R6)                          ;
                       1C   A6       57  90 00091          MOVB    R7, 28(R6)                                 ; 1608
                                         04 00095          RET                                                ; 1607
     08  A2        01       02       6A  F0 00096 7$:      INSV    FDL$GL_SWITCH, #2, #1, 8(R2)               ; 1611
                                         04 0009C          RET                                                ;
                            7E       68  3C 0009D 8$:      MOVZWL  FDL$AB_STRING, -(SP)                       ; 1617
                            6B       01  FB 000A0          CALLS   #1, FDL$$GET_VM                            ;
                       10   A2       50  D0 000A3          MOVL    R0, 16(R2)                                 ;
                            57       68  3C 000A7          MOVZWL  FDL$AB_STRING, R7                          ; 1619
                            50  04   A8  D0 000AA          MOVL    FDL$AB_STRING+4, R0                        ; 1620
                            56       69  D0 000AE          MOVL    JNL_XAB, R6                                ; 1621
           10  B6           60       57  28 000B1          MOVC3   R7, (R0), @16(R6)                          ;
                       0C   A6       57  90 000B6          MOVB    R7, 12(R6)                                 ; 1624
                                         04 000BA          RET                                                ; 1623
                            51  08   A2  9E 000BB 9$:      MOVAB   8(R2), R1                                  ; 1631
                            61       23  8A 000BF          BICB2   #35, (R1)                                  ; 1633
                            50 00000000G 00  D0 000C2      MOVL    FDL$GL_QUALIFIER, R0                       ; 1635
                            13       50  D1 000C9          CMPL    R0, #19                                    ;
                            04       12 000CC              BNEQ    10$                                        ;
                            61       02  88 000CE          BISB2   #2, (R1)                                   ; 1637
                                         04 000D1          RET                                                ;
                            14       50  D1 000D2 10$:     CMPL    R0, #20                                    ; 1639
                            04       12 000D5              BNEQ    11$                                        ;
                            61       01  88 000D7          BISB2   #1, (R1)                                   ; 1641
                                         04 000DA          RET                                                ;
                            15       50  D1 000DB 11$:     CMPL    R0, #21                                    ; 1643
                            03       12 000DE              BNEQ    12$                                        ;
                            61       20  88 000E0          BISB2   #32, (R1)                                  ; 1645
                                         04 000E3 12$:     RET                                                ; 1653

; Routine Size:  228 bytes,    Routine Base:  _FDL$CODE + 03E1
;  942        1654  1 %SBTTL 'SET_ACL_P'
;  943        1655  1 ROUTINE SET_ACL_P : NOVALUE =
;  944        1656  1 !++
;  945        1657  1 !
;  946        1658  1 ! Functional Description:
;  947        1659  1 !
;  948        1660  1 !       Fill in the blanks for the ACL xab
;  949        1661  1 !
;  950        1662  1 ! Calling Sequence:
;  951        1663  1 !
;  952        1664  1 !       set_acl_p()
;  953        1665  1 !
;  954        1666  1 ! Input Parameters:
;  955        1667  1 !       none
;  956        1668  1 !
;  957        1669  1 ! Implicit Inputs:
;  958        1670  1 !
;  959        1671  1 !       fdl$secondary   - Secondary code
;  960        1672  1 !
;  961        1673  1 ! Output Parameters:
;  962        1674  1 !       none
;  963        1675  1 !
;  964        1676  1 ! Implicit Outputs:
;  965        1677  1 !       none
;  966        1678  1 !
;  967        1679  1 ! Routine Value:
;  968        1680  1 !       none
;  969        1681  1 !
;  970        1682  1 ! Routines Called:
;  971        1683  1 !
;  972        1684  1 !       none
;  973        1685  1 !
;  974        1686  1 ! Side Effects:
;  975        1687  1 !       none
;  976        1688  1 !
;  977        1689  1 !--
;  978        1690  1
;  979        1691  2     BEGIN
;  980        1692  2
;  981        1693  2 ! nop until there exists an ACLXAB
;  982        1694  2
;  983        1695  2     RETURN
;  984        1696  2
;  985        1697  1     END;



                                                  0000 00000 SET_ACL_P:
                                                                    .WORD   Save nothing                                 ; 1655
                                                    04 00002        RET                                                  ; 1697

; Routine Size:  3 bytes,    Routine Base:  _FDL$CODE + 04C5
```

```
1 %SBTTL  'SET_FILE_P'
1 ROUTINE SET_FILE_P : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Fill in the blanks for the fab
1 !
1 ! Calling Sequence:
1 !
1 !       set_file_p()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       fdl$secondary   - Secondary code
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       SS$_NORMAL or error from set_prot
1 !
1 ! Routines Called:
1 !
1 !       fdl$$get_vm
1 !       set_prot
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     REGISTER
2         PARSED_FAB : REF BLOCK [ ,BYTE ];
2
2     PARSED_FAB = .FDL$AB_PARSED_FAB;
2
2     ! Set the fab according to the secondary parsed
2     !
2     SELECT .FDL$GL_SECONDARY OF
2     SET
2         [ FDL$C_ALL ]    : PARSED_FAB [ FAB$L_ALQ ] = .FDL$GL_NUMBER;
2
2         [ FDL$C_BKTUP ] : 0;
2
2         [ FDL$C_BTC ]    : PARSED_FAB [ FAB$V_CBT ] = .FDL$GL_SWITCH;
2
3         [ FDL$C_BKTSIZ ]: BEGIN
```

```

                                    PARSED_FAB [ FAB$B_BKS ] = .FDL$GL_NUMBER;

                                    ! Stuff the bucket size into the array for latter
                                    !
                                    FDL$AB_AREA_BKZ [ 0 ] - .FDL$GL_NUMBER

                                    END;

                [ FDL$C_CLUSIZ ]: 0;

                [ FDL$C_FCTX ]  : PARSED_FAB [ FAB$L_CTX ] = .FDL$GL_NUMBER;

                [ FDL$C_CONT ]  : PARSED_FAB [ FAB$V_CTG ] = .FDL$GL_SWITCH;

                [ FDL$C_CIF ]   : PARSED_FAB [ FAB$V_CIF ] = .FDL$GL_SWITCH;

                [ FDL$C_DFNAM ] : BEGIN

                                    ! Allocate a buffer for the string and copy it into it
                                    !
                                    PARSED_FAB [ FAB$L_DNA ] =
                                          FDL$$GET_VM( .FDL$AB_STRING [ DSC$W_LENGTH ] );

                                    CH$MOVE( .FDL$AB_STRING [ DSC$W_LENGTH ],
                                             .FDL$AB_STRING [ DSC$A_POINTER ],
                                             .PARSED_FAB [ FAB$L_DNA ] );

                                    PARSED_FAB [ FAB$B_DNS ] =
                                                      .FDL$AB_STRING [ DSC$W_LENGTH ]
                                    END;

                [ FDL$C_DEFWRT ] : PARSED_FAB [ FAB$V_DFW ] = .FDL$GL_SWITCH;

                [ FDL$C_DOC ]   : PARSED_FAB [ FAB$V_DLT ] = .FDL$GL_SWITCH;

                [ FDL$C_DIR ]   : PARSED_FAB [ FAB$V_TMP ] = .FDL$GL_SWITCH;

! not supported V4.0
!               [ FDL$C_EODEL ] : PARSED_FAB [ FAB$V_EDL ] = .FDL$GL_SWITCH;

                [ FDL$C_EXTEN ] : PARSED_FAB [ FAB$W_DEQ ] = .FDL$GL_NUMBER;

                [ FDL$C_GBC ]   : PARSED_FAB [ FAB$W_GBC ] = .FDL$GL_NUMBER;

                [ FDL$C_MTBLSIZ]: PARSED_FAB [ FAB$W_BLS ] = .FDL$GL_NUMBER;

                [ FDL$C_MTCP ]  : PARSED_FAB [ FAB$V_POS ] = .FDL$GL_SWITCH;

                [ FDL$C_MTNEF ] : PARSED_FAB [ FAB$V_NEF ] = .FDL$GL_SWITCH;

                [ FDL$C_MTPRO ] : SET_PROT();

                [ FDL$C_MTREW ] : PARSED_FAB [ FAB$V_RWO ] = .FDL$GL_SWITCH;

                [ FDL$C_MTRWC ] : PARSED_FAB [ FAB$V_RWC ] = .FDL$GL_SWITCH;

```

```
2        [ FDL$C_MAXRECN]: PARSED_FAB [ FAB$L_MRN ] = .FDL$GL_NUMBER;
2
2        [ FDL$C_MAXVER] : PARSED_FAB [ FAB$V_MXV ] = .FDL$GL_SWITCH;
2
3        [ FDL$C_NAME ]  : BEGIN
3                          ! Check for non-null name string
3                          !
3                          IF .FDL$AB_STRING [DSC$W_LENGTH] NEQ 0
3                          THEN
4                              BEGIN
4                                  ! Allocate a buffer for the string and copy it
4                                  !
4                                  PARSED_FAB [ FAB$L_FNA ] =
4                                          FDL$$GET_VM( .FDL$AB_STRING [ DSC$W_LENGTH ] );
4
4                                  CH$MOVE( .FDL$AB_STRING [ DSC$W_LENGTH ],
4                                           .FDL$AB_STRING [ DSC$A_POINTER ],
4                                           .PARSED_FAB [ FAB$L_FNA ] );
3                              END;
3                          PARSED_FAB [ FAB$B_FNS ] =
3                                              .FDL$AB_STRING [ DSC$W_LENGTH ]
3                          END;
2
2        [ FDL$C_NFS ]   : PARSED_FAB [ FAB$V_NFS ] = .FDL$GL_SWITCH;
2
2        [ FDL$C_ORG ]   : PARSED_FAB [ FAB$B_ORG ] = .FDL$GL_QUALIFIER;
2
2        [ FDL$C_OFP ]   : PARSED_FAB [ FAB$V_OFP ] = .FDL$GL_SWITCH;
2
2        [ FDL$C_OWNER ] : SET_PROT();
2
2        [ FDL$C_POC ]   : PARSED_FAB [ FAB$V_SPL ] = .FDL$GL_SWITCH;
2
2        [ FDL$C_PROT ]  : SET_PROT();
2
2        [ FDL$C_READC ] : PARSED_FAB [ FAB$V_RCK ] = .FDL$GL_SWITCH;
2
3        [ FDL$C_REVISN ]: BEGIN
3
3                          ! If the revision xab has not been connected then connect it
3                          !
3                          IF .REVISION_XAB EQLU 0
3                          THEN
3
3                              ! Allocate the xab an enter it into the chain
3                              !
3                              REVISION_XAB = ALLOCATE_XAB ( XAB$C_RDT, 0 );
3
3                          REVISION_XAB [ XAB$W_RVN ] = .FDL$GL_NUMBER
3
2                          END;
2
2        [ FDL$C_SQO ]   : PARSED_FAB [ FAB$V_SQO ] = .FDL$GL_SWITCH;
2
2        [ FDL$C_SOC ]   : PARSED_FAB [ FAB$V_SCF ] = .FDL$GL_SWITCH;
2
2        [ FDL$C_SUPER ] : PARSED_FAB [ FAB$V_SUP ] = .FDL$GL_SWITCH;
```

```
; 1158    1869  2
; 1159    1870  2             [ FDL$C_TEMPO ] : PARSED_FAB [ FAB$V_TMD ] = .FDL$GL_SWITCH;
; 1160    1871  2
; 1161    1872  2             [ FDL$C_TOC ]   : PARSED_FAB [ FAB$V_TEF ] = .FDL$GL_SWITCH;
; 1162    1873  2
; 1163    1874  2             [ FDL$C_UFO ]   : PARSED_FAB [ FAB$V_UFO ] = .FDL$GL_SWITCH;
; 1164    1875  2
; 1165    1876  2             [ FDL$C_WIN ]   : PARSED_FAB [ FAB$B_RTV ] = .FDL$GL_NUMBER;
; 1166    1877  2
; 1167    1878  2             [ FDL$C_WRITEC ]: PARSED_FAB [ FAB$V_WCK ] = .FDL$GL_SWITCH;
; 1168    1879  2
; 1169    1880  2         TES;
; 1170    1881  2
; 1171    1882  2         RETURN
; 1172    1883  2
; 1173    1884  1         END;



                                              0FFC 00000 SET_FILE_P:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11              ; 1699
                         5B 00000000G   00  9E 00002           MOVAB   FDL$AB_STRING, R11                                ;
                         5A 00000000G   00  9E 00009           MOVAB   FDL$GL_NUMBER, R10                                ;
                         59 00000000G   00  9E 00010           MOVAB   FDL$GL_SWITCH, R9                                 ;
                         56 00000000G   00  D0 00017           MOVL    FDL$AB_PARSED_FAB, PARSED_FAB                     ; 1742
                         57 00000000G   00  D0 0001E           MOVL    FDL$GL_SECONDARY, R7                              ; 1746
               00000048  8F             57  D1 00025           CMPL    R7, #72                                           ; 1748
                                        04  12 0002C           BNEQ    1$                                                ;
                     10  A6             6A  D0 0002E           MOVL    FDL$GL_NUMBER, 16(PARSED_FAB)                     ;
               00000049  8F             57  D1 00032 1$:       CMPL    R7, #73                                           ; 1752
                                        06  12 00039           BNEQ    2$                                                ;
  06   A6           01   05             69  F0 0003B           INSV    FDL$GL_SWITCH, #5, #1, 6(PARSED_FAB)              ;
               0000004A  8F             57  D1 00041 2$:       CMPL    R7, #74                                           ; 1754
                                        11  12 00048           BNEQ    3$                                                ;
                         50             6A  D0 0004A           MOVL    FDL$GL_NUMBER, R0                                 ; 1756
                     3E  A6             50  90 0004D           MOVB    R0, 62(PARSED_FAB)                                ;
                         51 00000000G   00  D0 00051           MOVL    FDL$AB_AREA_BKZ, R1                               ; 1760
                         61             50  90 00058           MOVB    R0, (R1)                                          ;
               0000004C  8F             57  D1 0005B 3$:       CMPL    R7, #76                                           ; 1766
                                        04  12 00062           BNEQ    4$                                                ;
                     18  A6             6A  D0 00064           MOVL    FDL$GL_NUMBER, 24(PARSED_FAB)                     ;
               0000004D  8F             57  D1 00068 4$:       CMPL    R7, #77                                           ; 1768
                                        06  12 0006F           BNEQ    5$                                                ;
  06   A6           01   04             69  F0 00071           INSV    FDL$GL_SWITCH, #4, #1, 6(PARSED_FAB)              ;
               0000004E  8F             57  D1 00077 5$:       CMPL    R7, #78                                           ; 1770
                                        06  12 0007E           BNEQ    6$                                                ;
  07   A6           01   01             69  F0 00080           INSV    FDL$GL_SWITCH, #1, #1, 7(PARSED_FAB)              ;
               0000004F  8F             57  D1 00086 6$:       CMPL    R7, #79                                           ; 1772
                                        1E  12 0008D           BNEQ    7$                                                ;
                         7E             6B  3C 0008F           MOVZWL  FDL$AB_STRING, -(SP)                              ; 1777
               00000000V 00             01  FB 00092           CALLS   #1, FDL$$GET_VM                                   ;
                     30  A6             50  D0 00099           MOVL    R0, 48(PARSED_FAB)                                ;
                         58             6B  3C 0009D           MOVZWL  FDL$AB_STRING, R8                                 ; 1779
                         50         04  AB  D0 000A0           MOVL    FDL$AB_STRING+4, R0                               ; 1780
            30   B6      60             58  28 000A4           MOVC3   R8, (R0), @48(PARSED_FAB)                         ; 1781
```

```
                              35    A6              58  90 000A9        MOVB    R8, 53(PARSED_FAB)                              ; 1784
                        00000050    8F              57  D1 000AD  7$:   CMPL    R7, #80                                         ; 1787
                                                    06  12 000B4        BNEQ    8$
    04     A6            01         05              69  F0 000B6        INSV    FDL$GL_SWITCH, #5, #1, 4(PARSED_FAB)
                        00000051    8F              57  D1 000BC  8$:   CMPL    R7, #81                                         ; 1789
                                                    06  12 000C3        BNEQ    9$
    05     A6            01         07              69  F0 000C5        INSV    FDL$GL_SWITCH, #7, #1, 5(PARSED_FAB)
                        00000052    8F              57  D1 000CB  9$:   CMPL    R7, #82                                         ; 1791
                                                    06  12 000D2        BNEQ    10$
    04     A6            01         03              69  F0 000D4        INSV    FDL$GL_SWITCH, #3, #1, 4(PARSED_FAB)
                        00000054    8F              57  D1 000DA  10$:  CMPL    R7, #84                                         ; 1796
                                                    04  12 000E1        BNEQ    11$
                              14    A6              6A  B0 000E3        MOVW    FDL$GL_NUMBER, 20(PARSED_FAB)
                        00000055    8F              57  D1 000E7  11$:  CMPL    R7, #85                                         ; 1798
                                                    04  12 000EE        BNEQ    12$
                              48    A6              6A  B0 000F0        MOVW    FDL$GL_NUMBER, 72(PARSED_FAB)
                        00000056    8F              57  D1 000F4  12$:  CMPL    R7, #86                                         ; 1800
                                                    04  12 000FB        BNEQ    13$
                              3C    A6              6A  B0 000FD        MOVW    FDL$GL_NUMBER, 60(PARSED_FAB)
                        00000057    8F              57  D1 00101  13$:  CMPL    R7, #87                                         ; 1802
                                                    06  12 00108        BNEQ    14$
    05     A6            01         00              69  F0 0010A        INSV    FDL$GL_SWITCH, #0, #1, 5(PARSED_FAB)
                        00000058    8F              57  D1 00110  14$:  CMPL    R7, #88                                         ; 1804
                                                    06  12 00117        BNEQ    15$
    05     A6            01         02              69  F0 00119        INSV    FDL$GL_SWITCH, #2, #1, 5(PARSED_FAB)
                        00000059    8F              57  D1 0011F  15$:  CMPL    R7, #89                                         ; 1806
                                                    07  12 00126        BNEQ    16$
                       00000000V    00              00  FB 00128        CALLS   #0, SET_PROT
                        0000005A    8F              57  D1 0012F  16$:  CMPL    R7, #90                                         ; 1808
                                                    06  12 00136        BNEQ    17$
    04     A6            01         07              69  F0 00138        INSV    FDL$GL_SWITCH, #7, #1, 4(PARSED_FAB)
                        0000005B    8F              57  D1 0013E  17$:  CMPL    R7, #91                                         ; 1810
                                                    06  12 00145        BNEQ    18$
    05     A6            01         03              69  F0 00147        INSV    FDL$GL_SWITCH, #3, #1, 5(PARSED_FAB)
                        0000005C    8F              57  D1 0014D  18$:  CMPL    R7, #92                                         ; 1812
                                                    04  12 00154        BNEQ    19$
                              38    A6              6A  D0 00156        MOVL    FDL$GL_NUMBER, 56(PARSED_FAB)
                        0000005D    8F              57  D1 0015A  19$:  CMPL    R7, #93                                         ; 1814
                                                    06  12 00161        BNEQ    20$
    04     A6            01         01              69  F0 00163        INSV    FDL$GL_SWITCH, #1, #1, 4(PARSED_FAB)
                        0000005E    8F              57  D1 00169  20$:  CMPL    R7, #94                                         ; 1816
                                                    1F  12 00170        BNEQ    22$
                                    50              6B  3C 00172        MOVZWL  FDL$AB_STRING, R0                               ; 1819
                                                    16  13 00175        BEQL    21$
                                                    50  DD 00177        PUSHL   R0                                              ; 1825
                       00000000V    00              01  FB 00179        CALLS   #1, FDL$$GET_VM
                              2C    A6              50  D0 00180        MOVL    R0, 44(PARSED_FAB)
                                    50        04    AB  D0 00184        MOVL    FDL$AB_STRING+4, R0                             ; 1828
                  2C     B6         60              6B  28 00188        MOVC3   FDL$AB_STRING, (R0), @44(PARSED_FAB)            ; 1829
                              34    A6              6B  90 0018D  21$:  MOVB    FDL$AB_STRING, 52(PARSED_FAB)                   ; 1832
                        00000060    8F              57  D1 00191  22$:  CMPL    R7, #96                                         ; 1835
                                                    06  12 00198        BNEQ    23$
    06     A6            01         00              69  F0 0019A        INSV    FDL$GL_SWITCH, #0, #1, 6(PARSED_FAB)
                        00000062    8F              57  D1 001A0  23$:  CMPL    R7, #98                                         ; 1837
                                                    08  12 001A7        BNEQ    24$
                              1D    A6  00000000G   00  90 001A9        MOVB    FDL$GL_QUALIFIER, 29(PARSED_FAB)
                        00000061    8F              57  D1 001B1  24$:  CMPL    R7, #97                                         ; 1839
```

```
                                                       06   12 001B8          BNEQ    25$                                         ;
    07   A6        01                       05         69   F0 001BA          INSV    FDL$GL_SWITCH, #5, #1, 7(PARSED_FAB)        ;
                        00000063            8F         57   D1 001C0 25$:     CMPL    R7, #99                                     ; 1841
                                                       07   12 001C7          BNEQ    26$                                         ;
                        0000000V            00         00   FB 001C9          CALLS   #0, SET_PROT                                ;
                        00000064            8F         57   D1 001D0 26$:     CMPL    R7, #100                                    ; 1843
                                                       06   12 001D7          BNEQ    27$                                         ;
    05   A6        01                       05         69   F0 001D9          INSV    FDL$GL_SWITCH, #5, #1, 5(PARSED_FAB)        ;
                        00000065            8F         57   D1 001DF 27$:     CMPL    R7, #101                                    ; 1845
                                                       07   12 001E6          BNEQ    28$                                         ;
                        0000000V            00         00   FB 001E8          CALLS   #0, SET_PROT                                ;
                        00000066            8F         57   D1 001EF 28$:     CMPL    R7, #102                                    ; 1847
                                                       06   12 001F6          BNEQ    29$                                         ;
    06   A6        01                       07         69   F0 001F8          INSV    FDL$GL_SWITCH, #7, #1, 6(PARSED_FAB)        ;
                        00000067            8F         57   D1 001FE 29$:     CMPL    R7, #103                                    ; 1849
                                                       24   12 00205          BNEQ    31$                                         ;
                                        00000000'  00  D5 00207               TSTL    REVISION_XAB                                ; 1853
                                                       11   12 0020D          BNEQ    30$                                         ;
                                            7E         1E   7D 0020F          MOVQ    #30, -(SP)                                  ; 1858
                        0000000V            00         02   FB 00212          CALLS   #2, ALLOCATE_XAB                            ;
                        00000000'           00         50   D0 00219          MOVL    R0, REVISION_XAB                            ;
                                            50  00000000'  00   D0 00220 30$: MOVL    REVISION_XAB, R0                            ; 1860
                              08            A0         6A   B0 00227          MOVW    FDL$GL_NUMBER, 8(R0)                        ;
                        00000068            8F         57   D1 0022B 31$:     CMPL    R7, #104                                    ; 1864
                                                       06   12 00232          BNEQ    32$                                         ;
    04   A6        01                       06         69   F0 00234          INSV    FDL$GL_SWITCH, #6, #1, 4(PARSED_FAB)        ;
                        00000069            8F         57   D1 0023A 32$:     CMPL    R7, #105                                    ; 1866
                                                       06   12 00241          BNEQ    33$                                         ;
    05   A6        01                       06         69   F0 00243          INSV    FDL$GL_SWITCH, #6, #1, 5(PARSED_FAB)        ;
                        0000006A            8F         57   D1 00249 33$:     CMPL    R7, #106                                    ; 1868
                                                       06   12 00250          BNEQ    34$                                         ;
    04   A6        01                       02         69   F0 00252          INSV    FDL$GL_SWITCH, #2, #1, 4(PARSED_FAB)        ;
                        0000006B            8F         57   D1 00258 34$:     CMPL    R7, #107                                    ; 1870
                                                       06   12 0025F          BNEQ    35$                                         ;
    04   A6        01                       04         69   F0 00261          INSV    FDL$GL_SWITCH, #4, #1, 4(PARSED_FAB)        ;
                        0000006C            8F         57   D1 00267 35$:     CMPL    R7, #108                                    ; 1872
                                                       06   12 0026E          BNEQ    36$                                         ;
    07   A6        01                       04         69   F0 00270          INSV    FDL$GL_SWITCH, #4, #1, 7(PARSED_FAB)        ;
                        0000006D            8F         57   D1 00276 36$:     CMPL    R7, #109                                    ; 1874
                                                       06   12 0027D          BNEQ    37$                                         ;
    06   A6        01                       01         69   F0 0027F          INSV    FDL$GL_SWITCH, #1, #1, 6(PARSED_FAB)        ;
                        0000006E            8F         57   D1 00285 37$:     CMPL    R7, #110                                    ; 1876
                                                       04   12 0028C          BNEQ    38$                                         ;
                              1C            A6         6A   90 0028E          MOVB    FDL$GL_NUMBER, 28(PARSED_FAB)               ;
                        0000006F            8F         57   D1 00292 38$:     CMPL    R7, #111                                    ; 1878
                                                       06   12 00299          BNEQ    39$                                         ;
    05   A6        01                       01         69   F0 0029B          INSV    FDL$GL_SWITCH, #1, #1, 5(PARSED_FAB)        ;
                                                            04 002A1 39$:     RET                                                 ; 1884

; Routine Size:  674 bytes,     Routine Base:  _FDL$CODE + 04C8
```

```
%SBTTL  'SET_KEY_P'
ROUTINE SET_KEY_P : NOVALUE =
!++
!
! Functional Description:
!
!       Fill in the blanks for the key xab
!
! Calling Sequence:
!
!       set_key_p()
!
! Input Parameters:
!       none
!
! Implicit Inputs:
!
!       fdl$secondary   - Secondary code
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!       none
!
! Routines Called:
!
!       allocate_xab
!
! Side Effects:
!       none
!
!--

    BEGIN

    ! Find out if there is a current xab  if not then get one
    !
    IF .CURRENT_XAB EQL 0
    THEN
        BEGIN

        ALLOCATE_XAB ( XAB$C_KEY, .FDL$GL_PRINUM );

        CURRENT_XAB [ XAB$B_REF ] = .FDL$GL_PRINUM

        END
    ELSE

        ! If the current xab is not the same type or number of what we want
        ! then get a new one
        !
        IF ( .CURRENT_XAB [ XAB$B_COD ] NEQ XAB$C_KEY ) OR
           ( .CURRENT_XAB [ XAB$B_REF ] NEQ .FDL$GL_PRINUM )
```

```
2              THEN
3                  BEGIN
3
3                  ALLOCATE_XAB ( XAB$C_KEY, .FDL$GL_PRINUM );
3
3                  CURRENT_XAB [ XAB$B_REF ] = .FDL$GL_PRINUM
3
2                  END;
2
2          ! Set the key xab fields
2          !
2          CASE .FDL$GL_SECONDARY FROM FDL$C_CHANGE TO FDL$C_SEGTYP OF
2          SET
2              [ FDL$C_CHANGE ]: CURRENT_XAB [ XAB$V_CHG ] = .FDL$GL_SWITCH;
2
2              [ FDL$C_DAREA ] : CURRENT_XAB [ XAB$B_DAN ] = .FDL$GL_NUMBER;
2
2              [ FDL$C_DFILL ] : CURRENT_XAB [ XAB$W_DFL ] = .FDL$GL_NUMBER;
2
2              [ FDL$C_DATKC ] : CURRENT_XAB [ XAB$V_KEY_NCMPR ] = NOT .FDL$GL_SWITCH;
2
2              [ FDL$C_DATRC ] : CURRENT_XAB [ XAB$V_DAT_NCMPR ] = NOT .FDL$GL_SWITCH;
2
2              [ FDL$C_DUPS ]  : CURRENT_XAB [ XAB$V_DUP ] = .FDL$GL_SWITCH;
2
2              [ FDL$C_IAREA ] : CURRENT_XAB [ XAB$B_IAN ] = .FDL$GL_NUMBER;
2
2              [ FDL$C_IDXC ]  : CURRENT_XAB [ XAB$V_IDX_NCMPR ] = NOT .FDL$GL_SWITCH;
2
2              [ FDL$C_IFILL ] : CURRENT_XAB [ XAB$W_IFL ] = .FDL$GL_NUMBER;
2
3              [ FDL$C_KYNAME ]: BEGIN
3                                CURRENT_XAB [ XAB$L_KNM ] = FDL$$GET_VM ( 32 );
3                                CH$COPY( .FDL$AB_STRING [ DSC$W_LENGTH ],
3                                         .FDL$AB_STRING [ DSC$A_POINTER ],
3                                         SPACE,32,
3                                         .CURRENT_XAB [ XAB$L_KNM ] )
2                                END;
2
2              [ FDL$C_LAREA ] : CURRENT_XAB [ XAB$B_LAN ] = .FDL$GL_NUMBER;
2
2              [ FDL$C_NULL ]  : CURRENT_XAB [ XAB$V_NUL ] = .FDL$GL_SWITCH;
2
2              [ FDL$C_NULLVAL]: CURRENT_XAB [ XAB$B_NUL ] = .FDL$GL_QUALIFIER;
2
2              [ FDL$C_PROL ]  : IF .CURRENT_XAB [ XAB$B_REF ] EQLU 0
2                                THEN
2                                    CURRENT_XAB [ XAB$B_PROLOG ] = .FDL$GL_NUMBER;
2
2              [ FDL$C_SEGLEN ]: CASE .FDL$GL_SECNUM FROM 0 TO 7 OF
2                                SET
2                                    [ 0 ] : CURRENT_XAB [ XAB$B_SIZ0 ] = .FDL$GL_NUMBER;
2                                    [ 1 ] : CURRENT_XAB [ XAB$B_SIZ1 ] = .FDL$GL_NUMBER;
2                                    [ 2 ] : CURRENT_XAB [ XAB$B_SIZ2 ] = .FDL$GL_NUMBER;
2                                    [ 3 ] : CURRENT_XAB [ XAB$B_SIZ3 ] = .FDL$GL_NUMBER;
2                                    [ 4 ] : CURRENT_XAB [ XAB$B_SIZ4 ] = .FDL$GL_NUMBER;
2                                    [ 5 ] : CURRENT_XAB [ XAB$B_SIZ5 ] = .FDL$GL_NUMBER;
```

```
; 1289      1999  2                      [ 6 ] : CURRENT_XAB [ XAB$B_SIZ6 ] = .FDL$GL_NUMBER;
; 1290      2000  2                      [ 7 ] : CURRENT_XAB [ XAB$B_SIZ7 ] = .FDL$GL_NUMBER;
; 1291      2001  2                 TES;
; 1292      2002  2
; 1293      2003  2      [ FDL$C_SEGPOS ]: CASE .FDL$GL_SECNUM FROM 0 TO 7 OF
; 1294      2004  2                 SET
; 1295      2005  2                      [ 0 ] : CURRENT_XAB [ XAB$W_POS0 ] = .FDL$GL_NUMBER;
; 1296      2006  2                      [ 1 ] : CURRENT_XAB [ XAB$W_POS1 ] = .FDL$GL_NUMBER;
; 1297      2007  2                      [ 2 ] : CURRENT_XAB [ XAB$W_POS2 ] = .FDL$GL_NUMBER;
; 1298      2008  2                      [ 3 ] : CURRENT_XAB [ XAB$W_POS3 ] = .FDL$GL_NUMBER;
; 1299      2009  2                      [ 4 ] : CURRENT_XAB [ XAB$W_POS4 ] = .FDL$GL_NUMBER;
; 1300      2010  2                      [ 5 ] : CURRENT_XAB [ XAB$W_POS5 ] = .FDL$GL_NUMBER;
; 1301      2011  2                      [ 6 ] : CURRENT_XAB [ XAB$W_POS6 ] = .FDL$GL_NUMBER;
; 1302      2012  2                      [ 7 ] : CURRENT_XAB [ XAB$W_POS7 ] = .FDL$GL_NUMBER;
; 1303      2013  2                 TES;
; 1304      2014  2
; 1305      2015  2      [ FDL$C_SEGTYP ]: CASE .FDL$GL_SECNUM FROM 0 TO 7 OF
; 1306      2016  2                 SET
; 1307      2017  3                      [ 0 ] : BEGIN
; 1308      2018  3                              CURRENT_XAB [ XAB$B_DTP ] = .FDL$GL_QUALIFIER;
; 1309      2019  3                              CURRENT_XAB [ XAB$B_TYP0 ] = .FDL$GL_QUALIFIER
; 1310      2020  2                              END;
; 1311      2021  2                      [ 1 ] : CURRENT_XAB [ XAB$B_TYP1 ] = .FDL$GL_QUALIFIER;
; 1312      2022  2                      [ 2 ] : CURRENT_XAB [ XAB$B_TYP2 ] = .FDL$GL_QUALIFIER;
; 1313      2023  2                      [ 3 ] : CURRENT_XAB [ XAB$B_TYP3 ] = .FDL$GL_QUALIFIER;
; 1314      2024  2                      [ 4 ] : CURRENT_XAB [ XAB$B_TYP4 ] = .FDL$GL_QUALIFIER;
; 1315      2025  2                      [ 5 ] : CURRENT_XAB [ XAB$B_TYP5 ] = .FDL$GL_QUALIFIER;
; 1316      2026  2                      [ 6 ] : CURRENT_XAB [ XAB$B_TYP6 ] = .FDL$GL_QUALIFIER;
; 1317      2027  2                      [ 7 ] : CURRENT_XAB [ XAB$B_TYP7 ] = .FDL$GL_QUALIFIER;
; 1318      2028  2                 TES;
; 1319      2029  2
; 1320      2030  2      TES;
; 1321      2031  2
; 1322      2032  2      RETURN
; 1323      2033  2
; 1324      2034  1      END;



                                      OFFC 00000 SET_KEY_P:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11              ; 1886
                        5B 00000000G   00  9E 00002      MOVAB   FDL$GL_QUALIFIER, R11                            ;
                        5A 00000000G   00  9E 00009      MOVAB   FDL$GL_SECNUM, R10                               ;
                        59 00000000G   00  9E 00010      MOVAB   FDL$GL_PRINUM, R9                                ;
                        58 00000000'   00  9E 00017      MOVAB   CURRENT_XAB, R8                                  ;
                        57 00000000G   00  9E 0001E      MOVAB   FDL$GL_SWITCH, R7                                ;
                        56 00000000G   00  9E 00025      MOVAB   FDL$GL_NUMBER, R6                                ;
                        52             68  D0 0002C      MOVL    CURRENT_XAB, R2                                  ; 1926
                                       0D  13 0002F      BEQL    1$                                               ;
                        15             62  91 00031      CMPB    (R2), #21                                        ; 1940
                                       08  12 00034      BNEQ    1$                                               ;
    69        17   A2   08             00  ED 00036      CMPZV   #0, #8, 23(R2), FDL$GL_PRINUM                    ; 1941
                                       12  13 0003C      BEQL    2$                                               ;
                                       69  DD 0003E 1$:  PUSHL   FDL$GL_PRINUM                                    ; 1945
                                       15  DD 00040      PUSHL   #21                                              ;
```

```
                    00000000V  00            02 FB 00042          CALLS   #2, ALLOCATE_XAB                                ;
                               50            68 D0 00049          MOVL    CURRENT_XAB, R0                                 ; 1947
                           17  A0            69 90 0004C          MOVB    FDL$GL_PRINUM, 23(R0)                           ;
                               52            68 D0 00050 2$:      MOVL    CURRENT_XAB, R2                                 ; 1955
                10 00000077    8F  00000000G 00 CF 00053          CASEL   FDL$GL_SECONDARY, #119, #16                     ; 1953
      0033          002E     0029          0022    0005F 3$:      .WORD   4$-3$,-                                         ;
      0053          004E     0047          003D    00067                  5$-3$,-                                         ;
      008A          0062     0085          005D    0006F                  6$-3$,-                                         ;
      00E0          00A1     0096          0091    00077                  7$-3$,-                                         ;
                                           011F    0007F                  8$-3$,-                                         ;
                                                                          9$-3$,-                                         ;
                                                                          10$-3$,-                                        ;
                                                                          11$-3$,-                                        ;
                                                                          12$-3$,-                                        ;
                                                                          14$-3$,-                                        ;
                                                                          13$-3$,-                                        ;
                                                                          15$-3$,-                                        ;
                                                                          16$-3$,-                                        ;
                                                                          17$-3$,-                                        ;
                                                                          19$-3$,-                                        ;
                                                                          29$-3$,-                                        ;
                                                                          39$-3$                                          ;
    12   A2         01         01            67 F0 00081 4$:      INSV    FDL$GL_SWITCH, #1, #1, 18(R2)                   ; 1955
                                             04 00087             RET                                                     ;
                           0A  A2            66 90 00088 5$:      MOVB    FDL$GL_NUMBER, 10(R2)                           ; 1957
                                             04 0008C             RET                                                     ;
                           1C  A2            66 B0 0008D 6$:      MOVW    FDL$GL_NUMBER, 28(R2)                           ; 1959
                                             04 00091             RET                                                     ;
                               50            67 D2 00092 7$:      MCOML   FDL$GL_SWITCH, R0                               ; 1961
    12   A2         01         06            50 F0 00095          INSV    R0, #6, #1, 18(R2)                              ;
                                             04 0009B             RET                                                     ;
                               50            67 D2 0009C 8$:      MCOML   FDL$GL_SWITCH, R0                               ; 1963
    12   A2         01         07            50 F0 0009F          INSV    R0, #7, #1, 18(R2)                              ;
                                             04 000A5             RET                                                     ;
    12   A2         01         00            67 F0 000A6 9$:      INSV    FDL$GL_SWITCH, #0, #1, 18(R2)                   ; 1965
                                             04 000AC             RET                                                     ;
                           08  A2            66 90 000AD 10$:     MOVB    FDL$GL_NUMBER, 8(R2)                            ; 1967
                                             04 000B1             RET                                                     ;
                               50            67 D2 000B2 11$:     MCOML   FDL$GL_SWITCH, R0                               ; 1969
    12   A2         01         03            50 F0 000B5          INSV    R0, #3, #1, 18(R2)                              ;
                                             04 000BB             RET                                                     ;
                           1A  A2            66 B0 000BC 12$:     MOVW    FDL$GL_NUMBER, 26(R2)                           ; 1971
                                             04 000C0             RET                                                     ;
                                             20 DD 000C1 13$:     PUSHL   #32                                             ; 1974
                    00000000V  00            01 FB 000C3          CALLS   #1, FDL$$GET_VM                                 ;
                           38  A2            50 D0 000CA          MOVL    R0, 56(R2)                                      ;
                               51  00000000G 00 D0 000CE          MOVL    FDL$AB_STRING+4, R1                             ; 1976
                               50            68 D0 000D5          MOVL    CURRENT_XAB, R0                                 ; 1978
         20         20         61  00000000G 00 2C 000D8          MOVC5   FDL$AB_STRING, (R1), #32, #32, @56(R0)          ;
                                          38    B0 000E1                                                                  ;
                                             04 000E3             RET                                                     ; 1975
                           09  A2            66 90 000E4 14$:     MOVB    FDL$GL_NUMBER, 9(R2)                            ; 1981
                                             04 000E8             RET                                                     ;
    12   A2         01         02            67 F0 000E9 15$:     INSV    FDL$GL_SWITCH, #2, #1, 18(R2)                   ; 1983
                                             04 000EF             RET                                                     ;
                           15  A2            6B 90 000F0 16$:     MOVB    FDL$GL_QUALIFIER, 21(R2)                        ; 1985
                                             04 000F4             RET                                                     ;
```

```
                                 17    A2   95 000F5 17$:     TSTB    23(R2)                                   ; 1987
                                       01   13 000F8          BEQL    18$                                      ;
                                            04 000FA          RET                                              ;
                           48    A2         66 90 000FB 18$:     MOVB    FDL$GL_NUMBER, 72(R2)                    ; 1989
                                            04 000FF          RET                                              ; 1987
                           50               66 D0 00100 19$:     MOVL    FDL$GL_NUMBER, R0                        ; 1993
            07             00               6A CF 00103          CASEL   FDL$GL_SECNUM, #0, #7                    ; 1991
 001F       001A           0015             0010    00107 20$:     .WORD   21$-20$,-                                ;
 0033       002E           0029             0024    0010F                  22$-20$,-                                ;
                                                                       23$-20$,-                                ;
                                                                       24$-20$,-                                ;
                                                                       25$-20$,-                                ;
                                                                       26$-20$,-                                ;
                                                                       27$-20$,-                                ;
                                                                       28$-20$                                  ;
                           2E    A2         50 90 00117 21$:     MOVB    R0, 46(R2)                               ; 1993
                                            04 0011B          RET                                              ;
                           2F    A2         50 90 0011C 22$:     MOVB    R0, 47(R2)                               ; 1994
                                            04 00120          RET                                              ;
                           30    A2         50 90 00121 23$:     MOVB    R0, 48(R2)                               ; 1995
                                            04 00125          RET                                              ;
                           31    A2         50 90 00126 24$:     MOVB    R0, 49(R2)                               ; 1996
                                            04 0012A          RET                                              ;
                           32    A2         50 90 0012B 25$:     MOVB    R0, 50(R2)                               ; 1997
                                            04 0012F          RET                                              ;
                           33    A2         50 90 00130 26$:     MOVB    R0, 51(R2)                               ; 1998
                                            04 00134          RET                                              ;
                           34    A2         50 90 00135 27$:     MOVB    R0, 52(R2)                               ; 1999
                                            04 00139          RET                                              ;
                           35    A2         50 90 0013A 28$:     MOVB    R0, 53(R2)                               ; 2000
                                            04 0013E          RET                                              ; 1991
                           50               66 D0 0013F 29$:     MOVL    FDL$GL_NUMBER, R0                        ; 2005
            07             00               6A CF 00142          CASEL   FDL$GL_SECNUM, #0, #7                    ; 2003
 001F       001A           0015             0010    00146 30$:     .WORD   31$-30$,-                                ;
 0033       002E           0029             0024    0014E                  32$-30$,-                                ;
                                                                       33$-30$,-                                ;
                                                                       34$-30$,-                                ;
                                                                       35$-30$,-                                ;
                                                                       36$-30$,-                                ;
                                                                       37$-30$,-                                ;
                                                                       38$-30$                                  ;
                           1E    A2         50 B0 00156 31$:     MOVW    R0, 30(R2)                               ; 2005
                                            04 0015A          RET                                              ;
                           20    A2         50 B0 0015B 32$:     MOVW    R0, 32(R2)                               ; 2006
                                            04 0015F          RET                                              ;
                           22    A2         50 B0 00160 33$:     MOVW    R0, 34(R2)                               ; 2007
                                            04 00164          RET                                              ;
                           24    A2         50 B0 00165 34$:     MOVW    R0, 36(R2)                               ; 2008
                                            04 00169          RET                                              ;
                           26    A2         50 B0 0016A 35$:     MOVW    R0, 38(R2)                               ; 2009
                                            04 0016E          RET                                              ;
                           28    A2         50 B0 0016F 36$:     MOVW    R0, 40(R2)                               ; 2010
                                            04 00173          RET                                              ;
                           2A    A2         50 B0 00174 37$:     MOVW    R0, 42(R2)                               ; 2011
                                            04 00178          RET                                              ;
                           2C    A2         50 B0 00179 38$:     MOVW    R0, 44(R2)                               ; 2012
                                            04 0017D          RET                                              ; 2003
                                 50          6B  D0 0017E 39$:    MOVL    FDL$GL_QUALIFIER, R0                        ; 2018
                     07          00          6A  CF 00181         CASEL   FDL$GL_SECNUM, #0, #7                       ; 2015
        0023       001E        0019        0010     00185 40$:    .WORD   41$-40$,-                                   ;
        0037       0032        002D        0028     0018D                 42$-40$,-                                   ;
                                                                          43$-40$,-                                   ;
                                                                          44$-40$,-                                   ;
                                                                          45$-40$,-                                   ;
                                                                          46$-40$,-                                   ;
                                                                          47$-40$,-                                   ;
                                                                          48$-40$                                     ;
                             13    A2        50  90 00195 41$:    MOVB    R0, 19(R2)                                  ; 2018
                             40    A2        50  90 00199         MOVB    R0, 64(R2)                                  ; 2019
                                                 04 0019D         RET                                                 ;
                             41    A2        50  90 0019E 42$:    MOVB    R0, 65(R2)                                  ; 2021
                                                 04 001A2         RET                                                 ;
                             42    A2        50  90 001A3 43$:    MOVB    R0, 66(R2)                                  ; 2022
                                                 04 001A7         RET                                                 ;
                             43    A2        50  90 001A8 44$:    MOVB    R0, 67(R2)                                  ; 2023
                                                 04 001AC         RET                                                 ;
                             44    A2        50  90 001AD 45$:    MOVB    R0, 68(R2)                                  ; 2024
                                                 04 001B1         RET                                                 ;
                             45    A2        50  90 001B2 46$:    MOVB    R0, 69(R2)                                  ; 2025
                                                 04 001B6         RET                                                 ;
                             46    A2        50  90 001B7 47$:    MOVB    R0, 70(R2)                                  ; 2026
                                                 04 001BB         RET                                                 ;
                             47    A2        50  90 001BC 48$:    MOVB    R0, 71(R2)                                  ; 2027
                                                 04 001C0         RET                                                 ; 2034

; Routine Size:  449 bytes,    Routine Base:  _FDL$CODE + 076A
```

```
1 %SBTTL  'SET_RECORD_P'
1 ROUTINE SET_RECORD_P : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Fill in the blanks for the fab fields concerning the record
1 !
1 ! Calling Sequence:
1 !
1 !       set_record_p()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       fdl$secondary    - Secondary code
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routines Called:
1 !       none
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     REGISTER
2         PARSED_FAB       : REF BLOCK [ ,BYTE ];
2
2     PARSED_FAB = .FDL$AB_PARSED_FAB;
2
2     ! Set em up
2     !
2     CASE .FDL$GL_SECONDARY FROM FDL$C_BLKSPN TO FDL$C_SIZE OF
2     SET
2         [ FDL$C_BLKSPN ]: PARSED_FAB [ FAB$V_BLK ] = NOT .FDL$GL_SWITCH;
2
2         [ FDL$C_CARCTRL]: CASE .FDL$GL_QUALIFIER FROM FDL$C_NONE TO FDL$C_PRINT OF
2                         SET
2                             ! We must clear the other flags while setting the one
2                             ! we want (without clearing BLK if set)
2                             !
2                             [ FDL$C_NONE ]  : PARSED_FAB [ FAB$B_RAT ] =
2                                               .PARSED_FAB [ FAB$B_RAT ] AND
2                                                            FAB$M_BLK;
```

```
; 1383      2092  2                          [ FDL$C_CR ]    : PARSED_FAB [ FAB$B_RAT ] =
; 1384      2093  3                                              ( .PARSED_FAB [ FAB$B_RAT ] AND
; 1385      2094  2                                                  FAB$M_BLK ) OR FAB$M_CR;
; 1386      2095  2                          [ FDL$C_FTN ]   : PARSED_FAB [ FAB$B_RAT ] =
; 1387      2096  3                                              ( .PARSED_FAB [ FAB$B_RAT ] AND
; 1388      2097  2                                                  FAB$M_BLK ) OR FAB$M_FTN;
; 1389      2098  2                          [ FDL$C_PRINT ] : PARSED_FAB [ FAB$B_RAT ] =
; 1390      2099  3                                              ( .PARSED_FAB [ FAB$B_RAT ] AND
; 1391      2100  2                                                  FAB$M_BLK ) OR FAB$M_PRN;
; 1392      2101  2                      TES;
; 1393      2102  2
; 1394      2103  2          [ FDL$C_VFCSIZ ]: PARSED_FAB [ FAB$B_FSZ ] = .FDL$GL_NUMBER;
; 1395      2104  2
; 1396      2105  2          [ FDL$C_FMT ]    : PARSED_FAB [ FAB$B_RFM ] = .FDL$GL_QUALIFIER;
; 1397      2106  2
; 1398      2107  2          [ FDL$C_SIZE ]   : PARSED_FAB [ FAB$W_MRS ] = .FDL$GL_NUMBER;
; 1399      2108  2      TES;
; 1400      2109  2
; 1401      2110  2      RETURN
; 1402      2111  2
; 1403      2112  1      END;



                                             000C 00000 SET_RECORD_P:
                                                                   .WORD   Save R2,R3                                          ; 2036
                             53 00000000G  00  9E 00002            MOVAB   FDL$GL_NUMBER, R3
                             52 00000000G  00  9E 00009            MOVAB   FDL$GL_QUALIFIER, R2                                 ;
                             50 00000000G  00  D0 00010            MOVL    FDL$AB_PARSED_FAB, PARSED_FAB                        ; 2076
                 04 00000088 8F 00000000G  00  CF 00017            CASEL   FDL$GL_SECONDARY, #136, #4                           ; 2080
      005F            005A         0018      000A   00023 1$:      .WORD   2$-1$,-                                              ;
                                             0064   0002B                  3$-1$,-                                              ;
                                                                           9$-1$,-                                              ;
                                                                           10$-1$,-                                             ;
                                                                           11$-1$                                               ;
                             51 00000000G  00  D2 0002D 2$:        MCOML   FDL$GL_SWITCH, R1                                    ; 2082
    1E   A0           01           03          51  F0 00034            INSV    R1, #3, #1, 30(PARSED_FAB)                           ;
                                                   04 0003A            RET                                                          ;
                             51          1E    A0  9E 0003B 3$:        MOVAB   30(PARSED_FAB), R1                                   ; 2089
                      03           08          62  CF 0003F            CASEL   FDL$GL_QUALIFIER, #8, #3                             ; 2084
      002B            001C         000D      0008   00043 4$:      .WORD   5$-4$,-                                              ;
                                                                           6$-4$,-                                              ;
                                                                           7$-4$,-                                              ;
                                                                           8$-4$                                                ;
                             61          F7    8F  8A 0004B 5$:        BICB2   #-9, (R1)                                            ; 2090
                                                   04 0004F            RET                                                          ; 2089
                             50                61  9A 00050 6$:        MOVZBL  (R1), R0                                             ; 2093
                             50 FFFFFFF7       8F  CA 00053            BICL2   #-9, R0                                              ;
                      61     50                02  89 0005A            BISB3   #2, R0, (R1)                                         ; 2094
                                                   04 0005E            RET                                                          ; 2092
                             50                61  9A 0005F 7$:        MOVZBL  (R1), R0                                             ; 2096
                             50 FFFFFFF7       8F  CA 00062            BICL2   #-9, R0                                              ;
                      61     50                01  89 00069            BISB3   #1, R0, (R1)                                         ; 2097
                                                   04 0006D            RET                                                          ; 2095
                             50                61  9A 0006E 8$:        MOVZBL  (R1), R0                                             ; 2099
```

```
                        50 FFFFFFF7     8F  CA 00071          BICL2   #-9, R0                                     ;
            61          50              04  89 00078          BISB3   #4, R0, (R1)                                ; 2100
                                            04 0007C          RET                                                 ; 2084
                  3F    A0              63  90 0007D 9$:      MOVB    FDL$GL_NUMBER, 63(PARSED_FAB)               ; 2103
                                            04 00081          RET                                                 ;
                  1F    A0              62  90 00082 10$:     MOVB    FDL$GL_QUALIFIER, 31(PARSED_FAB)            ; 2105
                                            04 00086          RET                                                 ;
                  36    A0              63  B0 00087 11$:     MOVW    FDL$GL_NUMBER, 54(PARSED_FAB)               ; 2107
                                            04 0008B          RET                                                 ; 2112

; Routine Size:  140 bytes,    Routine Base:  _FDL$CODE + 092B
```

```
%SBTTL  'SET_ACCESS_P'
ROUTINE SET_ACCESS_P : NOVALUE =
!++
!
! Functional Description:
!
!         Fill in the blanks for the fab fields concerning access mode
!
! Calling Sequence:
!
!         set_access_p()
!
! Input Parameters:
!         none
!
! Implicit Inputs:
!
!         fdl$secondary   - Secondary code
!
! Output Parameters:
!         none
!
! Implicit Outputs:
!         none
!
! Routine Value:
!         none
!
! Routines Called:
!         none
!
! Side Effects:
!         none
!
!--

    BEGIN

    REGISTER
        PARSED_FAB      : REF BLOCK [ ,BYTE ];

    PARSED_FAB = .FDL$AB_PARSED_FAB;

    ! Set em up
    !
    CASE .FDL$GL_SECONDARY FROM FDL$C_FACBIO TO FDL$C_FACUPD OF
    SET
        [ FDL$C_FACBIO ] : PARSED_FAB [ FAB$V_BIO ] = .FDL$GL_SWITCH;

        [ FDL$C_FACDEL ] : PARSED_FAB [ FAB$V_DEL ] = .FDL$GL_SWITCH;

        [ FDL$C_FACGET ] : PARSED_FAB [ FAB$V_GET ] = .FDL$GL_SWITCH;

        [ FDL$C_FACPUT ] : PARSED_FAB [ FAB$V_PUT ] = .FDL$GL_SWITCH;

        [ FDL$C_FACBRO ] : PARSED_FAB [ FAB$V_BRO ] = .FDL$GL_SWITCH;

```

```
; 1462      2170  2          [ FDL$C_FACTRN ] : PARSED_FAB [ FAB$V_TRN ] = .FDL$GL_SWITCH;
; 1463      2171  2
; 1464      2172  2          [ FDL$C_FACUPD ] : PARSED_FAB [ FAB$V_UPD ] = .FDL$GL_SWITCH;
; 1465      2173  2      TES;
; 1466      2174  2
; 1467      2175  2      RETURN
; 1468      2176  2
; 1469      2177  1      END;




                                             0000 00000 SET_ACCESS_P:
                                                               .WORD    Save nothing                                    ; 2114
                          50 00000000G  00  D0 00002           MOVL     FDL$AB_PARSED_FAB, PARSED_FAB                   ; 2154
                          51        16  A0  9E 00009           MOVAB    22(PARSED_FAB), R1                              ; 2160
                          50 00000000G  00  D0 0000D           MOVL     FDL$GL_SWITCH, R0                               ;
              06          01 00000000G  00  CF 00014           CASEL    FDL$GL_SECONDARY, #1, #6                        ; 2158
     0020   001A        0014          000E     0001C 1$:       .WORD    2$-1$,-                                         ;
            0032        002C          0026     00024                    3$-1$,-                                         ;
                                                                        4$-1$,-                                         ;
                                                                        5$-1$,-                                         ;
                                                                        6$-1$,-                                         ;
                                                                        7$-1$,-                                         ;
                                                                        8$-1$                                           ;
       61         01          05          50  F0 0002A 2$:     INSV     R0, #5, #1, (R1)                                ; 2160
                                              04 0002F         RET                                                      ;
       61         01          02          50  F0 00030 3$:     INSV     R0, #2, #1, (R1)                                ; 2162
                                              04 00035         RET                                                      ;
       61         01          01          50  F0 00036 4$:     INSV     R0, #1, #1, (R1)                                ; 2164
                                              04 0003B         RET                                                      ;
       61         01          00          50  F0 0003C 5$:     INSV     R0, #0, #1, (R1)                                ; 2166
                                              04 00041         RET                                                      ;
       61         01          06          50  F0 00042 6$:     INSV     R0, #6, #1, (R1)                                ; 2168
                                              04 00047         RET                                                      ;
       61         01          04          50  F0 00048 7$:     INSV     R0, #4, #1, (R1)                                ; 2170
                                              04 0004D         RET                                                      ;
       61         01          03          50  F0 0004E 8$:     INSV     R0, #3, #1, (R1)                                ; 2172
                                              04 00053         RET                                                      ; 2177

; Routine Size:  84 bytes,    Routine Base:  _FDL$CODE + 09B7
```

```
%SBTTL  'SET_SHARING_P'
ROUTINE SET_SHARING_P : NOVALUE =
!++
!
! Functional Description:
!
!       Fill in the blanks for the fab fields concerning sharing
!
! Calling Sequence:
!
!       set_sharing_p()
!
! Input Parameters:
!       none
!
! Implicit Inputs:
!
!       fdl$secondary   - Secondary code
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!       none
!
! Routines Called:
!       none
!
! Side Effects:
!       none
!
!--

    BEGIN

    REGISTER
        PARSED_FAB      : REF BLOCK [ ,BYTE ];

    PARSED_FAB = .FDL$AB_PARSED_FAB;

    ! Set em up
    !
    CASE .FDL$GL_SECONDARY FROM FDL$C_SHRDEL TO FDL$C_SHRUPI OF
    SET
        [ FDL$C_SHRDEL ] : PARSED_FAB [ FAB$V_SHRDEL ] = .FDL$GL_SWITCH;

        [ FDL$C_SHRGET ] : PARSED_FAB [ FAB$V_SHRGET ] = .FDL$GL_SWITCH;

        [ FDL$C_SHRMSE ] : PARSED_FAB [ FAB$V_MSE ] = .FDL$GL_SWITCH;

        [ FDL$C_SHRNIL ] : PARSED_FAB [ FAB$V_NIL ] = .FDL$GL_SWITCH;

        [ FDL$C_SHRPUT ] : PARSED_FAB [ FAB$V_SHRPUT ] = .FDL$GL_SWITCH;

```

```
; 1528          2235  2              [ FDL$C_SHRUPD ] : PARSED_FAB [ FAB$V_SHRUPD ] = .FDL$GL_SWITCH;
; 1529          2236  2
; 1530          2237  2              [ FDL$C_SHRUPI ] : PARSED_FAB [ FAB$V_UPI ] = .FDL$GL_SWITCH;
; 1531          2238  2          TES;
; 1532          2239  2
; 1533          2240  2          RETURN
; 1534          2241  2
; 1535          2242  1          END;



                                                   0000 00000 SET_SHARING_P:
                                                                        .WORD     Save nothing                                          ; 2179
                                   50 00000000G   00   D0 00002         MOVL      FDL$AB_PARSED_FAB, PARSED_FAB                         ; 2219
                                   51        17   A0   9E 00009         MOVAB     23(PARSED_FAB), R1                                    ; 2225
                                   50 00000000G   00   D0 0000D         MOVL      FDL$GL_SWITCH, R0
                      06 0000008D  8F 00000000G   00   CF 00014         CASEL     FDL$GL_SECONDARY, #141, #6                            ; 2223
        0020                001A          0014       000E    00020 1$:  .WORD     2$-1$,-                                               ;
                            0032          002C       0026    00028                3$-1$,-                                               ;
                                                                                  4$-1$,-                                               ;
                                                                                  5$-1$,-                                               ;
                                                                                  6$-1$,-                                               ;
                                                                                  7$-1$,-                                               ;
                                                                                  8$-1$                                                 ;
          61                  01            02            50  F0 0002E 2$:  INSV      R0, #2, #1, (R1)                                  ; 2225
                                                              04 00033      RET                                                         ;
          61                  01            01            50  F0 00034 3$:  INSV      R0, #1, #1, (R1)                                  ; 2227
                                                              04 00039      RET                                                         ;
          61                  01            04            50  F0 0003A 4$:  INSV      R0, #4, #1, (R1)                                  ; 2229
                                                              04 0003F      RET                                                         ;
          61                  01            05            50  F0 00040 5$:  INSV      R0, #5, #1, (R1)                                  ; 2231
                                                              04 00045      RET                                                         ;
          61                  01            00            50  F0 00046 6$:  INSV      R0, #0, #1, (R1)                                  ; 2233
                                                              04 0004B      RET                                                         ;
          61                  01            03            50  F0 0004C 7$:  INSV      R0, #3, #1, (R1)                                  ; 2235
                                                              04 00051      RET                                                         ;
          61                  01            06            50  F0 00052 8$:  INSV      R0, #6, #1, (R1)                                  ; 2237
                                                              04 00057      RET                                                         ; 2242

; Routine Size:  88 bytes,    Routine Base:  _FDL$CODE + 0A0B
```

```
1 %SBTTL  'SET_CONNECT_P'
1 ROUTINE SET_CONNECT_P : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !           Fill in the blanks for the Rab fields
1 !
1 ! Calling Sequence:
1 !
1 !           set_connect_p()
1 !
1 ! Input Parameters:
1 !           none
1 !
1 ! Implicit Inputs:
1 !
1 !           fdl$secondary    - Secondary code
1 !
1 ! Output Parameters:
1 !           none
1 !
1 ! Implicit Outputs:
1 !           none
1 !
1 ! Routine Value:
1 !           none
1 !
1 ! Routines Called:
1 !           none
1 !
1 ! Side Effects:
1 !           none
1 !
1 !--
1
2     BEGIN
2
2     REGISTER
2         PARSED_RAB        : REF BLOCK [ ,BYTE ];
2
2     PARSED_RAB = .FDL$AB_PARSED_RAB;
2
2     ! Set em up
2     !
2     CASE .FDL$GL_SECONDARY FROM FDL$C_ASY TO FDL$C_WBH OF
2     SET
2         [ FDL$C_ASY ]    : PARSED_RAB [ RAB$V_ASY ] = .FDL$GL_SWITCH;
2
2         [ FDL$C_BIO ]    : PARSED_RAB [ RAB$V_BIO ] = .FDL$GL_SWITCH;
2
2         [ FDL$C_BUCODE ] : PARSED_RAB [ RAB$L_BKT ] = .FDL$GL_NUMBER;
2
2         [ FDL$C_RCTX ]   : PARSED_RAB [ RAB$L_CTX ] = .FDL$GL_NUMBER;
2
2         [ FDL$C_EOF ]    : PARSED_RAB [ RAB$V_EOF ] = .FDL$GL_SWITCH;
2
```

```
        [ FDL$C_FLOA ]    : PARSED_RAB [ RAB$V_LOA ] = .FDL$GL_SWITCH;

        [ FDL$C_FDEL ]    : PARSED_RAB [ RAB$V_FDL ] = .FDL$GL_SWITCH;

        [ FDL$C_KGE ]     : PARSED_RAB [ RAB$V_KGE ] = .FDL$GL_SWITCH;

        [ FDL$C_KGT ]     : PARSED_RAB [ RAB$V_KGT ] = .FDL$GL_SWITCH;

        [ FDL$C_KLIM ]    : PARSED_RAB [ RAB$V_LIM ] = .FDL$GL_SWITCH;

        [ FDL$C_KRF ]     : PARSED_RAB [ RAB$B_KRF ] = .FDL$GL_NUMBER;

        [ FDL$C_LOCMODE ] : PARSED_RAB [ RAB$V_LOC ] = .FDL$GL_SWITCH;

        [ FDL$C_REA ]     : PARSED_RAB [ RAB$V_REA ] = .FDL$GL_SWITCH;

        [ FDL$C_RLK ]     : PARSED_RAB [ RAB$V_RLK ] = .FDL$GL_SWITCH;

        [ FDL$C_ULK ]     : PARSED_RAB [ RAB$V_ULK ] = .FDL$GL_SWITCH;

        [ FDL$C_MBC ]     : PARSED_RAB [ RAB$B_MBC ] = .FDL$GL_NUMBER;

        [ FDL$C_MBF ]     : PARSED_RAB [ RAB$B_MBF ] = .FDL$GL_NUMBER;

        [ FDL$C_NLK ]     : PARSED_RAB [ RAB$V_NLK ] = .FDL$GL_SWITCH;

        [ FDL$C_NXR ]     : PARSED_RAB [ RAB$V_NXR ] = .FDL$GL_SWITCH;

        [ FDL$C_RAH ]     : PARSED_RAB [ RAB$V_RAH ] = .FDL$GL_SWITCH;

        [ FDL$C_RRL ]     : PARSED_RAB [ RAB$V_RRL ] = .FDL$GL_SWITCH;

        [ FDL$C_TMO ]     : PARSED_RAB [ RAB$B_TMO ] = .FDL$GL_NUMBER;

        [ FDL$C_TMENB ]   : PARSED_RAB [ RAB$V_TMO ] = .FDL$GL_SWITCH;

        [ FDL$C_TPT ]     : PARSED_RAB [ RAB$V_TPT ] = .FDL$GL_SWITCH;

        [ FDL$C_TTCCO ]   : PARSED_RAB [ RAB$V_CCO ] = .FDL$GL_SWITCH;

        [ FDL$C_TTCVT ]   : PARSED_RAB [ RAB$V_CVT ] = .FDL$GL_SWITCH;

        [ FDL$C_TTPMT ]   : PARSED_RAB [ RAB$V_PMT ] = .FDL$GL_SWITCH;

        [ FDL$C_TTPTA ]   : PARSED_RAB [ RAB$V_PTA ] = .FDL$GL_SWITCH;

        [ FDL$C_TTRNE ]   : PARSED_RAB [ RAB$V_RNE ] = .FDL$GL_SWITCH;

        [ FDL$C_TTRNF ]   : PARSED_RAB [ RAB$V_RNF ] = .FDL$GL_SWITCH;

        [ FDL$C_UIF ]     : PARSED_RAB [ RAB$V_UIF ] = .FDL$GL_SWITCH;

        [ FDL$C_WAT ]     : PARSED_RAB [ RAB$V_WAT ] = .FDL$GL_SWITCH;

        [ FDL$C_WBH ]     : PARSED_RAB [ RAB$V_WBH ] = .FDL$GL_SWITCH;
    TES;
```

```
; 1651          2357  2       RETURN
; 1652          2358  2
; 1653          2359  1       END;




                                                          000C 00000 SET_CONNECT_P:
                                                                             .WORD    Save R2,R3                                  ; 2244
                                      53 00000000G   00  9E 00002           MOVAB    FDL$GL_NUMBER, R3                          ;
                                      52 00000000G   00  9E 00009           MOVAB    FDL$GL_SWITCH, R2                          ;
                                      50 00000000G   00  D0 00010           MOVL     FDL$AB_PARSED_RAB, PARSED_RAB              ; 2284
                        20            23 00000000G   00  CF 00017           CASEL    FDL$GL_SECONDARY, #35, #32                 ; 2288
      0055              0050          0049          0042      0001F 1$:     .WORD    2$-1$,-                                    ;
      0084              0068          0061          005A      00027                  3$-1$,-                                    ;
      0089              007D          0076          006F      0002F                  4$-1$,-                                    ;
      00A5              009E          0097          0090      00037                  5$-1$,-                                    ;
      00BD              00B6          00AF          00AA      0003F                  6$-1$,-                                    ;
      00D7              00CB          00D0          00C4      00047                  7$-1$,-                                    ;
      00F3              00EC          00E5          00DE      0004F                  8$-1$,-                                    ;
      010F              0108          0101          00FA      00057                  12$-1$,-                                   ;
                                                    0116      0005F                  9$-1$,-                                    ;
                                                                                     10$-1$,-                                   ;
                                                                                     11$-1$,-                                   ;
                                                                                     13$-1$,-                                   ;
                                                                                     14$-1$,-                                   ;
                                                                                     15$-1$,-                                   ;
                                                                                     16$-1$,-                                   ;
                                                                                     17$-1$,-                                   ;
                                                                                     18$-1$,-                                   ;
                                                                                     19$-1$,-                                   ;
                                                                                     20$-1$,-                                   ;
                                                                                     21$-1$,-                                   ;
                                                                                     22$-1$,-                                   ;
                                                                                     24$-1$,-                                   ;
                                                                                     23$-1$,-                                   ;
                                                                                     25$-1$,-                                   ;
                                                                                     26$-1$,-                                   ;
                                                                                     27$-1$,-                                   ;
                                                                                     28$-1$,-                                   ;
                                                                                     29$-1$,-                                   ;
                                                                                     30$-1$,-                                   ;
                                                                                     31$-1$,-                                   ;
                                                                                     32$-1$,-                                   ;
                                                                                     33$-1$,-                                   ;
                                                                                     34$-1$                                     ;
    04   A0             01            00            62  F0 00061 2$:        INSV     FDL$GL_SWITCH, #0, #1, 4(PARSED_RAB)       ; 2290
                                                        04 00067            RET                                                 ;
    05   A0             01            03            62  F0 00068 3$:        INSV     FDL$GL_SWITCH, #3, #1, 5(PARSED_RAB)       ; 2292
                                                        04 0006E            RET                                                 ;
                                 38   A0            63  D0 0006F 4$:        MOVL     FDL$GL_NUMBER, 56(PARSED_RAB)              ; 2294
                                                        04 00073            RET                                                 ;
                                 18   A0            63  D0 00074 5$:        MOVL     FDL$GL_NUMBER, 24(PARSED_RAB)              ; 2296
                                                        04 00078            RET                                                 ;
    05   A0             01            00            62  F0 00079 6$:        INSV     FDL$GL_SWITCH, #0, #1, 5(PARSED_RAB)       ; 2298
                                                        04 0007F            RET                                                 ;
```

```
   05  A0        01         05      62  F0 00080 7$:    INSV   FDL$GL_SWITCH, #5, #1, 5(PARSED_RAB)    ; 2300
                                        04 00086        RET
   04  A0        01         06      62  F0 00087 8$:    INSV   FDL$GL_SWITCH, #6, #1, 4(PARSED_RAB)    ; 2302
                                        04 0008D        RET
   06  A0        01         05      62  F0 0008E 9$:    INSV   FDL$GL_SWITCH, #5, #1, 6(PARSED_RAB)    ; 2304
                                        04 00094        RET
   06  A0        01         06      62  F0 00095 10$:   INSV   FDL$GL_SWITCH, #6, #1, 6(PARSED_RAB)    ; 2306
                                        04 0009B        RET
   05  A0        01         06      62  F0 0009C 11$:   INSV   FDL$GL_SWITCH, #6, #1, 5(PARSED_RAB)    ; 2308
                                        04 000A2        RET
                       35   A0      63  90 000A3 12$:   MOVB   FDL$GL_NUMBER, 53(PARSED_RAB)           ; 2310
                                        04 000A7        RET
   06  A0        01         00      62  F0 000A8 13$:   INSV   FDL$GL_SWITCH, #0, #1, 6(PARSED_RAB)    ; 2312
                                        04 000AE        RET
   04  A0        01         02      62  F0 000AF 14$:   INSV   FDL$GL_SWITCH, #2, #1, 4(PARSED_RAB)    ; 2314
                                        04 000B5        RET
   06  A0        01         03      62  F0 000B6 15$:   INSV   FDL$GL_SWITCH, #3, #1, 6(PARSED_RAB)    ; 2316
                                        04 000BC        RET
   06  A0        01         02      62  F0 000BD 16$:   INSV   FDL$GL_SWITCH, #2, #1, 6(PARSED_RAB)    ; 2318
                                        04 000C3        RET
                       37   A0      63  90 000C4 17$:   MOVB   FDL$GL_NUMBER, 55(PARSED_RAB)           ; 2320
                                        04 000C8        RET
                       36   A0      63  90 000C9 18$:   MOVB   FDL$GL_NUMBER, 54(PARSED_RAB)           ; 2322
                                        04 000CD        RET
   06  A0        01         04      62  F0 000CE 19$:   INSV   FDL$GL_SWITCH, #4, #1, 6(PARSED_RAB)    ; 2324
                                        04 000D4        RET
   06  A0        01         07      62  F0 000D5 20$:   INSV   FDL$GL_SWITCH, #7, #1, 6(PARSED_RAB)    ; 2326
                                        04 000DB        RET
   05  A0        01         01      62  F0 000DC 21$:   INSV   FDL$GL_SWITCH, #1, #1, 5(PARSED_RAB)    ; 2328
                                        04 000E2        RET
   04  A0        01         03      62  F0 000E3 22$:   INSV   FDL$GL_SWITCH, #3, #1, 4(PARSED_RAB)    ; 2330
                                        04 000E9        RET
                       1F   A0      63  90 000EA 23$:   MOVB   FDL$GL_NUMBER, 31(PARSED_RAB)           ; 2332
                                        04 000EE        RET
   07  A0        01         01      62  F0 000EF 24$:   INSV   FDL$GL_SWITCH, #1, #1, 7(PARSED_RAB)    ; 2334
                                        04 000F5        RET
   04  A0        01         01      62  F0 000F6 25$:   INSV   FDL$GL_SWITCH, #1, #1, 4(PARSED_RAB)    ; 2336
                                        04 000FC        RET
   07  A0        01         07      62  F0 000FD 26$:   INSV   FDL$GL_SWITCH, #7, #1, 7(PARSED_RAB)    ; 2338
                                        04 00103        RET
   07  A0        01         02      62  F0 00104 27$:   INSV   FDL$GL_SWITCH, #2, #1, 7(PARSED_RAB)    ; 2340
                                        04 0010A        RET
   07  A0        01         06      62  F0 0010B 28$:   INSV   FDL$GL_SWITCH, #6, #1, 7(PARSED_RAB)    ; 2342
                                        04 00111        RET
   07  A0        01         05      62  F0 00112 29$:   INSV   FDL$GL_SWITCH, #5, #1, 7(PARSED_RAB)    ; 2344
                                        04 00118        RET
   07  A0        01         00      62  F0 00119 30$:   INSV   FDL$GL_SWITCH, #0, #1, 7(PARSED_RAB)    ; 2346
                                        04 0011F        RET
   07  A0        01         03      62  F0 00120 31$:   INSV   FDL$GL_SWITCH, #3, #1, 7(PARSED_RAB)    ; 2348
                                        04 00126        RET
   04  A0        01         04      62  F0 00127 32$:   INSV   FDL$GL_SWITCH, #4, #1, 4(PARSED_RAB)    ; 2350
                                        04 0012D        RET
   06  A0        01         01      62  F0 0012E 33$:   INSV   FDL$GL_SWITCH, #1, #1, 6(PARSED_RAB)    ; 2352
                                        04 00134        RET
   05  A0        01         02      62  F0 00135 34$:   INSV   FDL$GL_SWITCH, #2, #1, 5(PARSED_RAB)    ; 2354
                                        04 0013B        RET                                            ; 2359
```

```
; Routine Size:  316 bytes,    Routine Base:  _FDL$CODE + 0A63
```

```
1 %SBTTL  'SET PROT'
1 ROUTINE SET_PROT : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Fill in the blanks for the protection xab
1 !
1 ! Calling Sequence:
1 !
1 !       set_prot()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !
1 !       fdl$secondary   - Secondary code
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routines Called:
1 !       none
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     ! See if the protection xab has been allocated yet
2     !
2     IF .PROTECTION_XAB EQLU 0
2     THEN
2
2         ! Allocate the xab an enter it into the chain
2         !
2         PROTECTION_XAB = ALLOCATE_XAB ( XAB$C_PRO, 0 );
2
2     ! Set the fields according to the secondary
2     !
2     SELECTONEU .FDL$GL_SECONDARY OF
2     SET
2         [ FDL$C_MTPRO ] : PROTECTION_XAB [ XAB$B_MTACC ] = .FDL$GL_QUALIFIER;
2
2         [ FDL$C_PROT ]  : PROTECTION_XAB [ XAB$W_PRO ] = NOT .FDL$GL_PROTECTION;
2
2         [ FDL$C_OWNER ] : PROTECTION_XAB [ XAB$L_UIC ] = .FDL$GL_OWNER_UIC;
2     TES;
```

```
; 1712          2417  2
; 1713          2418  2         RETURN
; 1714          2419  2
; 1715          2420  1         END;



                                               0004 00000 SET_PROT:
                                                                   .WORD   Save R2                                    ; 2361
                        52 00000000'   00   9E 00002               MOVAB   PROTECTION_XAB, R2
                                       62   D5 00009               TSTL    PROTECTION_XAB                             ; 2400
                                       0D   12 0000B               BNEQ    1$
                        7E             13   7D 0000D               MOVQ    #19, -(SP)                                 ; 2405
              00000000V 00             02   FB 00010               CALLS   #2, ALLOCATE_XAB                           ;
                        62             50   D0 00017               MOVL    R0, PROTECTION_XAB                         ;
                        50 00000000G   00   D0 0001A 1$:           MOVL    FDL$GL_SECONDARY, R0                       ; 2409
              00000059  8F             50   D1 00021               CMPL    R0, #89                                    ; 2411
                                       0C   12 00028               BNEQ    2$                                         ;
                        50             62   D0 0002A               MOVL    PROTECTION_XAB, R0                         ;
                   0A   A0 00000000G   00   90 0002D               MOVB    FDL$GL_QUALIFIER, 10(R0)                   ;
                                            04 00035               RET                                                ;
              00000065  8F             50   D1 00036 2$:           CMPL    R0, #101                                   ; 2413
                                       0C   12 0003D               BNEQ    3$                                         ;
                        50             62   D0 0003F               MOVL    PROTECTION_XAB, R0                         ;
                   08   A0 00000000G   00   B2 00042               MCOMW   FDL$GL_PROTECTION, 8(R0)                   ;
                                            04 0004A               RET                                                ;
              00000063  8F             50   D1 0004B 3$:           CMPL    R0, #99                                    ; 2415
                                       0B   12 00052               BNEQ    4$                                         ;
                        50             62   D0 00054               MOVL    PROTECTION_XAB, R0                         ;
                   0C   A0 00000000G   00   D0 00057               MOVL    FDL$GL_OWNER_UIC, 12(R0)                   ;
                                            04 0005F 4$:           RET                                                ; 2420

; Routine Size:  96 bytes,    Routine Base:  _FDL$CODE + 0B9F
```

```
1 %SBTTL  'ALLOCATE_XAB'
1 ROUTINE ALLOCATE_XAB ( XAB_TYPE, XAB_NUM ) =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Allocates an RMS extended attribute block from virtual memory
1 !
1 !       ****************************************************************
1 !
1 !       NOTE: THIS ROUTINE ASSUMES XABs ARE CONNECTED TO THE $FAB !!!
1 !       IT WILL NOT WORK WITH XABs THAT ARE CONNECTED TO THE $RAB !!!
1 !
1 !       ****************************************************************
1 !
1 ! Calling Sequence:
1 !
1 !       allocate_xab( xab_type, xab_num )
1 !
1 ! Input Parameters:
1 !
1 !       xab_type        - The RMS code for the type of xab wanted ie. XAB$C_xab
1 !       xab_num         - Which xab is desired (for key and area xabs)
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       Pointer to the new xab (also pointed to by current xab)
1 !
1 ! Routines Called:
1 !
1 !       fdl$$get_vm
1 !
1 ! Side Effects:
1 !
1 !       current_xab pointes to the new xab
1 !
1 !--
1
2    BEGIN
2
2    LOCAL
2       XAB      : REF BLOCK [ ,BYTE ],
2       FOUND,
2       XAB_LEN,
2       NEW_XAB;
2
2     ! Find the size of the type of xab we want.
2     !
```

```
3        XAB_LEN = ( SELECTONEU .XAB_TYPE OF
3                    SET
3                    [ XAB$C_ALL ] : XAB$C_ALLLEN;
3                    [ XAB$C_DAT ] : XAB$C_DATLEN;
3                    [ XAB$C_JNL ] : XAB$C_JNLLEN;
3                    [ XAB$C_KEY ] : XAB$C_KEYLEN;
3                    [ XAB$C_PRO ] : XAB$C_PROLEN;
3                    [ XAB$C_RDT ] : XAB$C_RDTLEN;
2                    TES );
2
2        FOUND = _CLEAR;
2
2        ! See if the xab we need already exists
2        ! (if we're in the second parse)
2        !
3        IF (
4        ( .FDL$AB_CTRL [ FDL$V_REPARSE ] )
3        AND
4        ( ( .XAB_TYPE EQLU XAB$C_ALL ) OR ( .XAB_TYPE EQLU XAB$C_KEY ) )
2        ) THEN
3            BEGIN
3
3            XAB = .FDL$AB_PARSED_FAB [ FAB$L_XAB ];
3
3            WHILE .XAB NEQU 0
3            DO
4                BEGIN
4
5                IF (
7                    (( .XAB_TYPE EQLU XAB$C_ALL )
6                    AND
7                    ( .XAB [ XAB$B_COD ] EQLU XAB$C_ALL )
6                    AND
6                    ( .XAB [ XAB$B_AID ] EQLU .XAB_NUM ))
5                OR
7                    (( .XAB_TYPE EQLU XAB$C_KEY )
6                    AND
7                    ( .XAB [ XAB$B_COD ] EQLU XAB$C_KEY )
6                    AND
6                    ( .XAB [ XAB$B_REF ] EQLU .XAB_NUM ))
4                ) THEN
5                    BEGIN
5
5                    NEW_XAB = .XAB;
5                    FOUND = _SET;
5                    EXITLOOP;
5
4                    END;
4
4                XAB = .XAB [ XAB$L_NXT ];
4
3                END;
3
2            END;
2
2        IF NOT .FOUND
2        THEN
```

```
; 1831      2535  3            BEGIN
; 1832      2536  3
; 1833      2537  3            ! Allocate a buffer for the new xab
; 1834      2538  3            !
; 1835      2539  3            NEW_XAB = FDL$$GET_VM( .XAB_LEN );
; 1836      2540  3
; 1837      2541  3            ! If this is the first xab link it to the fab else just connect it to
; 1838      2542  3            ! the last xab in the chain
; 1839      2543  3            !
; 1840      2544  3            IF .FDL$AB_PARSED_FAB [ FAB$L_XAB ] EQL 0
; 1841      2545  3            THEN
; 1842      2546  3                FDL$AB_PARSED_FAB [ FAB$L_XAB ] = .NEW_XAB
; 1843      2547  3            ELSE
; 1844      2548  3                END_XAB [ XAB$L_NXT ] = .NEW_XAB;
; 1845      2549  3
; 1846      2550  3            END_XAB = .NEW_XAB;
; 1847      2551  3
; 1848      2552  2            END;
; 1849      2553  2
; 1850      2554  2        ! Make this xab the current one
; 1851      2555  2        !
; 1852      2556  2        CURRENT_XAB = .NEW_XAB;
; 1853      2557  2
; 1854      2558  2        IF NOT .FOUND
; 1855      2559  2        THEN
; 1856      2560  3            BEGIN
; 1857      2561  3
; 1858      2562  3            ! Init. some stuff in it
; 1859      2563  3            !
; 1860      2564  3            CURRENT_XAB [ XAB$B_COD ] = .XAB_TYPE;
; 1861      2565  3            CURRENT_XAB [ XAB$B_BLN ] = .XAB_LEN;
; 1862      2566  3            CURRENT_XAB [ XAB$L_NXT ] = 0;
; 1863      2567  3
; 1864      2568  2            END;
; 1865      2569  2
; 1866      2570  2        RETURN .CURRENT_XAB
; 1867      2571  2
; 1868      2572  1        END;
```

```
                              007C 00000 ALLOCATE_XAB:
                                                   .WORD    Save R2,R3,R4,R5,R6                        ; 2422
          56 00000000G   00   9E 00002             MOVAB    FDL$AB_PARSED_FAB, R6                      ;
          55 00000000'   00   9E 00009             MOVAB    CURRENT_XAB, R5                            ;
          52         04  AC   D0 00010             MOVL     XAB_TYPE, R2                               ; 2478
          14             52   D1 00014             CMPL     R2, #20                                    ; 2480
                         05   12 00017             BNEQ     1$                                         ;
          53             20   D0 00019             MOVL     #32, XAB_LEN                               ;
                         37   11 0001C             BRB      7$                                         ;
          12             52   D1 0001E 1$:         CMPL     R2, #18                                    ; 2481
                         05   12 00021             BNEQ     2$                                         ;
          53             2C   D0 00023             MOVL     #44, XAB_LEN                               ;
                         2D   11 00026             BRB      7$                                         ;
          22             52   D1 00028 2$:         CMPL     R2, #34                                    ; 2482
```

```
                                          05  12 0002B          BNEQ    3$                                    ;
                              53          3C  D0 0002D          MOVL    #60, XAB_LEN                          ;
                                          23  11 00030          BRB     7$                                    ;
                              15          52  D1 00032  3$:     CMPL    R2, #21                               ; 2483
                                          06  12 00035          BNEQ    4$                                    ;
                              53     4C   8F  9A 00037          MOVZBL  #76, XAB_LEN                          ;
                                          18  11 0003B          BRB     7$                                    ;
                              13          52  D1 0003D  4$:     CMPL    R2, #19                               ; 2484
                                          06  12 00040          BNEQ    5$                                    ;
                              53     58   8F  9A 00042          MOVZBL  #88, XAB_LEN                          ;
                                          0D  11 00046          BRB     7$                                    ;
                              1E          52  D1 00048  5$:     CMPL    R2, #30                               ; 2485
                                          05  13 0004B          BEQL    6$                                    ;
                              53          01  CE 0004D          MNEGL   #1, XAB_LEN                           ;
                                          03  11 00050          BRB     7$                                    ;
                              53          14  D0 00052  6$:     MOVL    #20, XAB_LEN                          ;
                                          54  D4 00055  7$:     CLRL    FOUND                                 ; 2488
                              47 00000000G 00 E9 00057          BLBC    FDL$AB_CTRL+2, 13$                    ; 2494
                              14          52  D1 0005E          CMPL    R2, #20                               ; 2496
                                          05  13 00061          BEQL    8$                                    ;
                              15          52  D1 00063          CMPL    R2, #21                               ;
                                          3D  12 00066          BNEQ    13$                                   ;
                              50          66  D0 00068  8$:     MOVL    FDL$AB_PARSED_FAB, R0                 ; 2500
                              51     24   A0  D0 0006B          MOVL    36(R0), XAB                           ;
                                          34  13 0006F  9$:     BEQL    13$                                   ; 2502
                              14          52  D1 00071          CMPL    R2, #20                               ; 2507
                                          0E  12 00074          BNEQ    10$                                   ;
                              14          61  91 00076          CMPB    (XAB), #20                            ; 2509
                                          09  12 00079          BNEQ    10$                                   ;
    08   AC      17   A1      08          00  ED 0007B          CMPZV   #0, #8, 23(XAB), XAB_NUM              ; 2511
                                          13  13 00082          BEQL    11$                                   ;
                              15          52  D1 00084  10$:    CMPL    R2, #21                               ; 2513
                                          16  12 00087          BNEQ    12$                                   ;
                              15          61  91 00089          CMPB    (XAB), #21                            ; 2515
                                          11  12 0008C          BNEQ    12$                                   ;
    08   AC      17   A1      08          00  ED 0008E          CMPZV   #0, #8, 23(XAB), XAB_NUM              ; 2517
                                          08  12 00095          BNEQ    12$                                   ;
                              50          51  D0 00097  11$:    MOVL    XAB, NEW_XAB                          ; 2521
                              54          01  D0 0009A          MOVL    #1, FOUND                             ; 2522
                                          06  11 0009D          BRB     13$                                   ; 2519
                              51     04   A1  D0 0009F  12$:    MOVL    4(XAB), XAB                           ; 2527
                                          CA  11 000A3          BRB     9$                                    ; 2502
                              23          54  E8 000A5  13$:    BLBS    FOUND, 16$                            ; 2533
                                          53  DD 000A8          PUSHL   XAB_LEN                               ; 2539
                   00000000V  00          01  FB 000AA          CALLS   #1, FDL$$GET_VM                       ;
                              51          66  D0 000B1          MOVL    FDL$AB_PARSED_FAB, R1                 ; 2544
                                     24   A1  D5 000B4          TSTL    36(R1)                                ;
                                          06  12 000B7          BNEQ    14$                                   ;
                         24   A1          50  D0 000B9          MOVL    NEW_XAB, 36(R1)                       ; 2546
                                          08  11 000BD          BRB     15$                                   ;
                              51     04   A5  D0 000BF  14$:    MOVL    END_XAB, R1                           ; 2548
                         04   A1          50  D0 000C3          MOVL    NEW_XAB, 4(R1)                        ;
                         04   A5          50  D0 000C7  15$:    MOVL    NEW_XAB, END_XAB                      ; 2550
                              65          50  D0 000CB  16$:    MOVL    NEW_XAB, CURRENT_XAB                  ; 2556
                              0D          54  E8 000CE          BLBS    FOUND, 17$                            ; 2558
                              50          65  D0 000D1          MOVL    CURRENT_XAB, R0                       ; 2564
                              60          52  90 000D4          MOVB    R2, (R0)                              ;
```

```
                01   A0          53  90 000D7          MOVB    XAB_LEN, 1(R0)                              ; 2565
                           04    A0  D4 000DB          CLRL    4(R0)                                       ; 2566
                     50          65  D0 000DE  17$:    MOVL    CURRENT_XAB, R0                             ; 2570
                                     04 000E1          RET                                                 ; 2572

; Routine Size:  226 bytes,    Routine Base:  _FDL$CODE + 0BFF
```

```
%SBTTL  'FIND_ID'
ROUTINE FIND_ID : NOVALUE =
!++
!
! Functional Description:
!
!         Finds a file ID of a file specified by the FDL$STRING descriptor
!
! Calling Sequence:
!
!         find_id()
!
! Input Parameters:
!         none
!
! Implicit Inputs:
!         none
!
! Output Parameters:
!         none
!
! Implicit Outputs:
!         none
!
! Routine Value:
!         none
!
! Routines Called:
!
!         fdl$$get_vm
!
! Side Effects:
!         none
!
!--

   BEGIN

   LOCAL
        FAB     : REF BLOCK [ ,BYTE ],
        NAM     : REF BLOCK [ ,BYTE ];

   ! Get the address space for the FAB and the Name block
   !
   FAB = FDL$$GET_VM( FAB$K_BLN );

   NAM = FDL$$GET_VM( NAM$K_BLN + ESA_BUF_SIZ );

   !   +--------------+
   !   I    nam blk   I
   !   +--------------+
   !   I  exp str buf I
   !   +--------------+
   !
   ! Init the blocks and fill in all of the good stuff
   !
   $FAB_INIT ( FAB = .FAB,
; 1927      P 2630  2               FNA = .FDL$AB_STRING [ DSC$A_POINTER ],
; 1928      P 2631  2               FNS = .FDL$AB_STRING [ DSC$W_LENGTH ],
; 1929        2632  2               NAM = .NAM );
; 1930        2633  2
; 1931      P 2634  2       $NAM_INIT ( ESA = .NAM + NAM$K_BLN,
; 1932      P 2635  2                   ESS = ESA_BUF_SIZ,
; 1933        2636  2                   NAM = .NAM );
; 1934        2637  2
; 1935        2638  2       ! Parse and search for the file
; 1936        2639  2       !
; 1937        2640  3       IF $PARSE( FAB=.FAB )
; 1938        2641  2       THEN
; 1939        2642  2
; 1940        2643  3           IF $SEARCH( FAB=.FAB )
; 1941        2644  2           THEN
; 1942        2645  3               BEGIN
; 1943        2646  3
; 1944        2647  3               ! Get the old file ID
; 1945        2648  3               !
; 1946        2649  3               FDL$GL_FID1 = .NAM [ NAM$W_FID_NUM ];
; 1947        2650  3               FDL$GL_FID2 = .NAM [ NAM$W_FID_SEQ ];
; 1948        2651  3               FDL$GL_FID3 = .NAM [ NAM$W_FID_RVN ]
; 1949        2652  3
; 1950        2653  3               END
; 1951        2654  2           ELSE
; 1952        2655  2               SIGNAL( FDL$_RFLOC )
; 1953        2656  2       ELSE
; 1954        2657  2           SIGNAL( FDL$_RFLOC );
; 1955        2658  2
; 1956        2659  2       ! Deallocate the space we used
; 1957        2660  2       !
; 1958        2661  2       FDL$$FREE_VM( FAB$K_BLN, .FAB );
; 1959        2662  2       FDL$$FREE_VM( NAM$K_BLN+ESA_BUF_SIZ, .NAM );
; 1960        2663  2
; 1961        2664  2       RETURN
; 1962        2665  2
; 1963        2666  1       END;


                                                              .EXTRN  SYS$PARSE, SYS$SEARCH

                                           03FC 00000 FIND_ID:.WORD   Save R2,R3,R4,R5,R6,R7,R8,R9          ; 2574
                         59 00000000V   00  9E 00002          MOVAB   FDL$$GET_VM, R9                       :
                         58 00000000V   00  9E 00009          MOVAB   FDL$$FREE_VM, R8                      :
                         7E        50   8F  9A 00010          MOVZBL  #80, -(SP)                            ; 2617
                         69             01  FB 00014          CALLS   #1, FDL$$GET_VM                       :
                         57             50  D0 00017          MOVL    R0, FAB                               :
                         7E      015F   8F  3C 0001A          MOVZWL  #351, -(SP)                           ; 2619
                         69             01  FB 0001F          CALLS   #1, FDL$$GET_VM                       :
                         56             50  D0 00022          MOVL    R0, NAM                               :
   0050    8F         00 6E             00  2C 00025          MOVC5   #0, (SP), #0, #80, (FAB)              ; 2632
                                        67     0002C                                                        :
                         67      5003   8F  B0 0002D          MOVW    #20483, (FAB)                         :
                    16   A7             02  90 00032          MOVB    #2, 22(FAB)                           :
                    1F   A7             02  90 00036          MOVB    #2, 31(FAB)                           :
                    28   A7             56  D0 0003A          MOVL    NAM, 40(FAB)                          :
                        2C    A7 00000000G   00  D0 0003E         MOVL    FDL$AB_STRING+4, 44(FAB)              ;
                        34    A7 00000000G   00  90 00046         MOVB    FDL$AB_STRING, 52(FAB)                ;
  0060   8F          00       6E             00  2C 0004E         MOVC5   #0, (SP), #0, #96, (NAM)              ; 2636
                              66                    00055                                                       ;
                              66      6002   8F  B0 00056         MOVW    #24578, (NAM)                         ;
                        0A    A6             01  8E 0005B         MNEGB   #1, 10(NAM)                           ;
                        0C    A6        60   A6  9E 0005F         MOVAB   96(R6), 12(NAM)                       ;
                                             57  DD 00064         PUSHL   FAB                                   ; 2640
                 00000000G    00             01  FB 00066         CALLS   #1, SYS$PARSE                         ;
                              26             50  E9 0006D         BLBC    R0, 1$                                ;
                                             57  DD 00070         PUSHL   FAB                                   ; 2643
                 00000000G    00             01  FB 00072         CALLS   #1, SYS$SEARCH                        ;
                              1A             50  E9 00079         BLBC    R0, 1$                                ;
                 00000000G    00        24   A6  3C 0007C         MOVZWL  36(NAM), FDL$GL_FID1                  ; 2649
                 00000000G    00        26   A6  3C 00084         MOVZWL  38(NAM), FDL$GL_FID2                  ; 2650
                 00000000G    00        28   A6  3C 0008C         MOVZWL  40(NAM), FDL$GL_FID3                  ; 2651
                                             0D  11 00094         BRB     2$                                    ;
                                00000000G    8F  DD 00096 1$:     PUSHL   #FDL$_RFLOC                           ; 2657
                 00000000G    00             01  FB 0009C         CALLS   #1, LIB$SIGNAL                        ;
                                             57  DD 000A3 2$:     PUSHL   FAB                                   ; 2661
                              7E        50   8F  9A 000A5         MOVZBL  #80, -(SP)                            ;
                              68             02  FB 000A9         CALLS   #2, FDL$$FREE_VM                      ;
                                             56  DD 000AC         PUSHL   NAM                                   ; 2662
                              7E      015F   8F  3C 000AE         MOVZWL  #351, -(SP)                           ;
                              68             02  FB 000B3         CALLS   #2, FDL$$FREE_VM                      ;
                                                 04 000B6         RET                                           ; 2666

; Routine Size:  183 bytes,    Routine Base:  _FDL$CODE + 0CE1
```

```
%SBTTL  'GET_VM'
GLOBAL ROUTINE FDL$$GET_VM( BYTES ) =
!++
!
! Functional Description:
!
!       Allocate virtual memory and zeros it
!
! Calling Sequence:
!
!       fdl$$get_vm( bytes )
!
! Input Parameters:
!
!       bytes   - number of bytes to allocate
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!
!       address of the start of the buffer
!
! Routine Called:
!
!       lib$get_vm
!
! Side Effects:
!       none
!
!--

    BEGIN

    LOCAL
        VM_POINTER;

    ! If we don't succede signal an error and stop
    !
    IF NOT LIB$GET_VM ( BYTES,VM_POINTER )
    THEN
        SIGNAL_STOP ( FDL$_INSVIRMEM );

    ! Zero this address space
    !
    CH$FILL ( 0,.BYTES,.VM_POINTER );

    RETURN .VM_POINTER

    END;
```

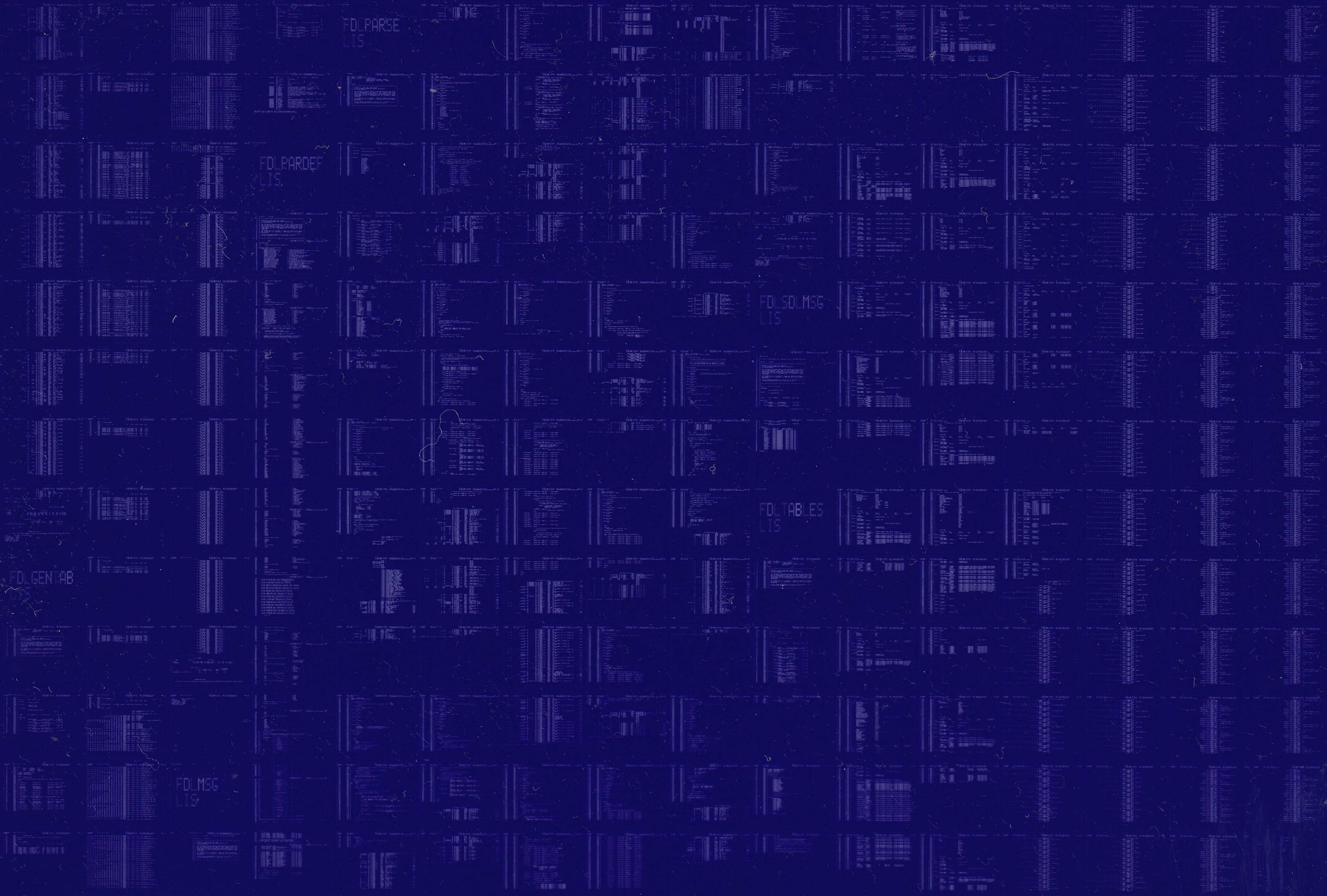
```
                                            003C 00000       .ENTRY  FDL$$GET_VM, Save R2,R3,R4,R5                         ; 2668
                        5E              04   C2 00002       SUBL2   #4, SP
                                        5E   DD 00005       PUSHL   SP                                                      ; 2712
                                   04   AC   9F 00007       PUSHAB  BYTES
                 00000000G  00          02   FB 0000A       CALLS   #2, LIB$GET_VM
                            0D          50   E8 00011       BLBS    R0, 1$
                              00000000G 8F   DD 00014       PUSHL   #FDL$_INSVIRMEM                                         ; 2714
                 00000000G  00          01   FB 0001A       CALLS   #1, LIB$STOP
   04   AC          00      6E          00   2C 00021  1$:  MOVC5   #0, (SP), #0, BYTES, @VM_POINTER                        ; 2718
                                   00   BE      00027
                        50              6E   D0 00029       MOVL    VM_POINTER, R0                                          ; 2720
                                             04 0002C       RET                                                             ; 2722

; Routine Size:  45 bytes,    Routine Base:  _FDL$CODE + 0D98
```

```
1 %SBTTL  'FREE_VM'
1 GLOBAL ROUTINE FDL$$FREE_VM( BYTES,ADDR ) : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Deallocate virtual memory
1 !
1 ! Calling Sequence:
1 !
1 !       fdl$$free_vm( bytes,addr )
1 !
1 ! Input Parameters:
1 !
1 !       bytes   - number of bytes to deallocate
1 !       addr    - address of block
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routine Called:
1 !
1 !       lib$free_vm
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         STATUS;
2
2     ! If we don't succede signal an error and stop
2     !
3     IF NOT ( STATUS = LIB$FREE_VM ( BYTES,ADDR ) )
2     THEN
2         SIGNAL_STOP ( .STATUS );
2
2     RETURN
2
1     END;
```

```
                                  0000 00000        .ENTRY  FDL$$FREE_VM, Save nothing                    : 2724
                       08  AC  9F 00002        PUSHAB  ADDR                                          : 2768
                       04  AC  9F 00005        PUSHAB  BYTES                                         :
        00000000G  00      02  FB 00008        CALLS   #2, LIB$FREE_VM                               :
                   09      50  E8 0000F        BLBS    STATUS, 1$                                    :
                           50  DD 00012        PUSHL   STATUS                                        : 2770
        00000000G  00      01  FB 00014        CALLS   #1, LIB$STOP                                  :
                           04  0001B 1$:       RET                                                   : 2774

; Routine Size:  28 bytes,    Routine Base:  _FDL$CODE + 0DC5


; 2074           2775  1
; 2075           2776  0 END ELUDOM



                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                                PSECT SUMMARY
;
;      Name                     Bytes                    Attributes
;
;  _FDL$OWN                        28  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _FDL$CODE                     3553  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                         Library Statistics
;
;                                         -------- Symbols --------       Pages       Processing
;      File                               Total   Loaded   Percent        Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1       9776     244         2         581           00:01.0



;                                COMMAND QUALIFIERS

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

; Size:          3553 code + 28 data bytes
; Run Time:         00:59.3
; Elapsed Time:     03:08.7
; Lines/CPU Min:     2809
; Lexemes/CPU-Min: 21493
; Memory Used:  276 pages
; Compilation Complete
```

0177 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



FDLPARSE
LIS

FDLPARDEF
LIS

FDLSDLMSG
LIS

FDLTABLES
LIS

FDLGENTAB
LIS

FDLMSG
LIS