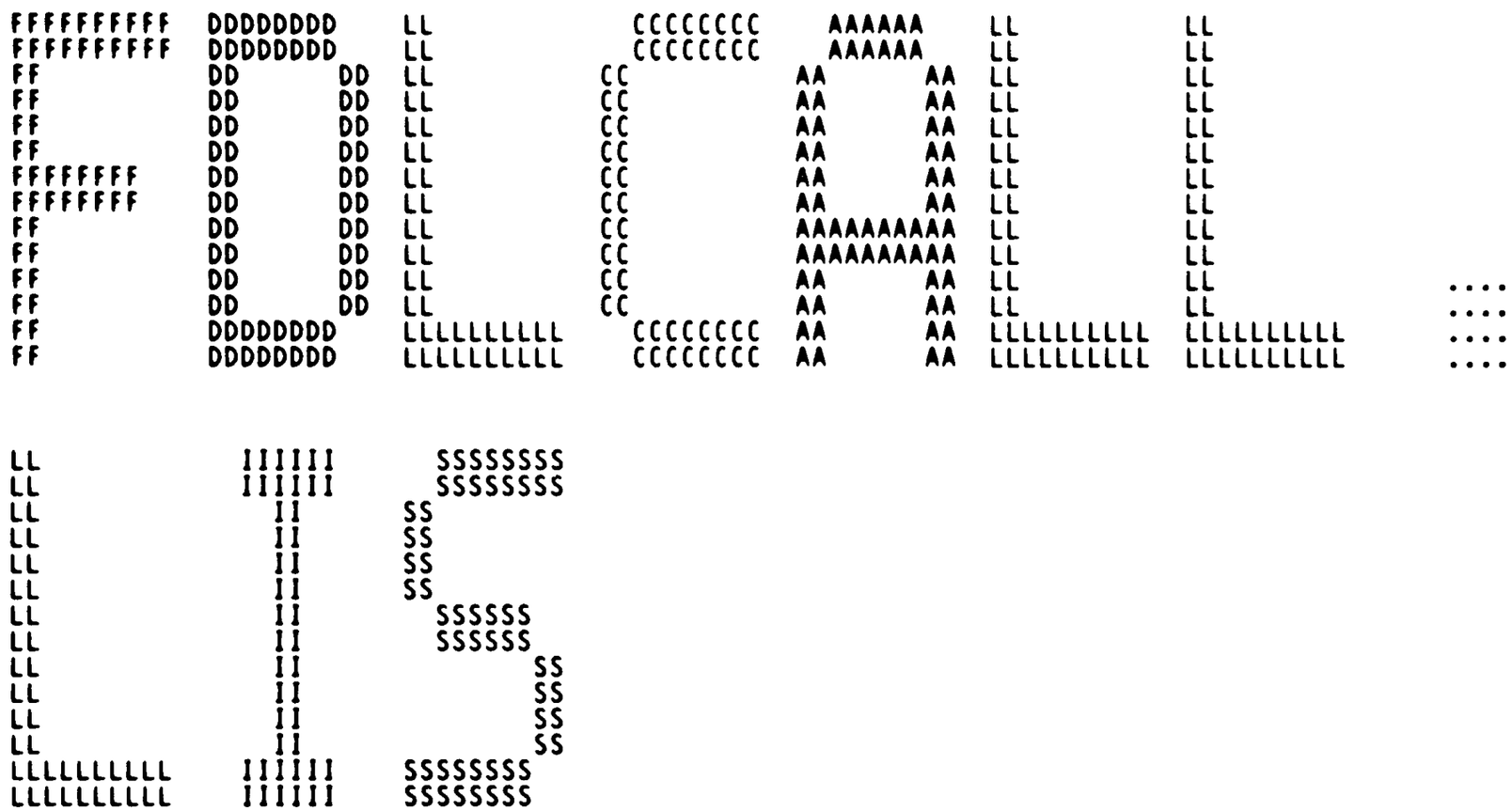
```
FFFFFFFFFFFFFFF  DDDDDDDDDDD      LLL
FFFFFFFFFFFFFFF  DDDDDDDDDDDD     LLL
FFFFFFFFFFFFFFF  DDDDDDDDDDDD     LLL
FFF              DDD          DDD LLL
FFF              DDD          DDD LLL
FFF              DDD          DDD LLL
FFF              DDD          DDD LLL
FFF              DDD          DDD 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      LLLLLLLLLLLLLLLL
FFF              DDDDDDDDDDDD      LLLLLLLLLLLLLLLL
FFF              DDDDDDDDDDDD      LLLLLLLLLLLLLLLL
```

FILEID**FDLCALL



```
;  1          0001  0 %TITLE  'VAX-11 FDL Utilities'
;  2          0002  0 MODULE  FDLCALL ( IDENT='V04-000',
;  3          0003  0                   ADDRESSING_MODE ( EXTERNAL = GENERAL ),
;  4          0004  0                   ADDRESSING_MODE ( NONEXTERNAL = GENERAL ),
;  5          0005  0                   OPTLEVEL=3
;  6          0006  0                   ) =
;  7          0007  0
;  8          0008  1 BEGIN
;  9          0009  1
;  10         0010  1 !**************************************************************************
;  11         0011  1 !*                                                                        *
;  12         0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;  13         0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;  14         0014  1 !*  ALL RIGHTS RESERVED.                                                   *
;  15         0015  1 !*                                                                        *
;  16         0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  17         0017  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;  18         0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  19         0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  20         0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  21         0021  1 !*  TRANSFERRED.                                                           *
;  22         0022  1 !*                                                                        *
;  23         0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  24         0024  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  25         0025  1 !*  CORPORATION.                                                           *
;  26         0026  1 !*                                                                        *
;  27         0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  28         0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;  29         0029  1 !*                                                                        *
;  30         0030  1 !*                                                                        *
;  31         0031  1 !**************************************************************************
```

```
;    33          0032  1 !++
;    34          0033  1 !
;    35          0034  1 ! Facility:     VAX-11 FDL Utilities
;    36          0035  1 !
;    37          0036  1 ! Abstract:
;    38          0037  1 !               Callable routines
;    39          0038  1 !
;    40          0039  1 ! Contents:
;    41          0040  1 !               FDL$CREATE
;    42          0041  1 !               FDL$PARSE
;    43          0042  1 !               FDL$RELEASE
;    44          0043  1 !               FDL$GENERATE
;    45          0044  1 !               HANDLER
;    46          0045  1 !
;    47          0046  1 ! Environment:
;    48          0047  1 !
;    49          0048  1 !               VAX/VMS Operating System
;    50          0049  1 !
;    51          0050  1 !--
;    52          0051  1
;    53          0052  1 !
;    54          0053  1 ! Author:       Keith B Thompson        Creation Date   June-81
;    55          0054  1 !
;    56          0055  1 !
;    57          0056  1 ! Modified by:
;    58          0057  1 !
;    59          0058  1 !               V03-015 JWT0192         Jim Teague              2-Aug-1984
;    60          0059  1 !                       RRB0019 clears the "valid stmt flag" on entry to all
;    61          0060  1 !                       routines in this module.  When FDL$CREATE calls
;    62          0061  1 !                       FDL$PARSE without a statement number argument, the
;    63          0062  1 !                       flag is cleared and remains off on return to
;    64          0063  1 !                       FDL$CREATE, even if it was specified on entry to
;    65          0064  1 !                       FDL$CREATE.
;    66          0065  1 !
;    67          0066  1 !               V03-014 RRB0019         Rowland R. Bradley      17-Apr-1984
;    68          0067  1 !                       Clear valid statement flag upon entry to all major
;    69          0068  1 !                       entry routines.  This fixes an elusive bug which
;    70          0069  1 !                       appears on multiple calls to the these routines.
;    71          0070  1 !
;    72          0071  1 !               V03-013 RAS0290         Ron Schaefer            10-Apr-1984
;    73          0072  1 !                       Fix interface to FDL$GENERATE so that the output
;    74          0073  1 !                       string descriptor/return length are processed
;    75          0074  1 !                       correctly for return FDL strings.
;    76          0075  1 !
;    77          0076  1 !               V03-012 KFH0012         Ken Henderson            8 Oct 1983
;    78          0077  1 !                       Recomment call to FDL$RELEASE
;    79          0078  1 !                       in FDL$CREATE until nasty bug
;    80          0079  1 !                       can be fixed in FDL$RELEASE!
;    81          0080  1 !
;    82          0081  1 !               V03-011 KFH0011         Ken Henderson           23 Aug 1983
;    83          0082  1 !                       Uncomment call to FDL$RELEASE
;    84          0083  1 !                       in FDL$CREATE.
;    85          0084  1 !                       Fix calls to GET_VM and FREE_VM.
;    86          0085  1 !                       Fix accvio if FDL$GL_PCALL is 0.
;    87          0086  1 !
;    88          0087  1 !               V03-010 KFH0010         Ken Henderson           29 Jul 1983
;    89          0088  1 !                       Fixed arg checking in FDL$GENERATE
```

```
;    90          0089  1 !
;    91          0090  1 !               Check status of calls to LIB$...
;    92          0091  1 !
;    93          0092  1 !       V03-009 KFH0009         Ken Henderson           26 Apr 1983
;    94          0093  1 !               Add [retlen] to FDL$GENERATE call.
;    95          0094  1 !
;    96          0095  1 !       V03-008 KFH0008         Ken Henderson           14 Apr 1983
;    97          0096  1 !               Add [retlen],[sts],[stv] parameters
;    98          0097  1 !               to FDL$CREATE call.
;    99          0098  1 !
;   100          0099  1 !       V03-007 KFH0007         Ken Henderson           22 Mar 1983
;   101          0100  1 !               Temporarily comment out the call to
;   102          0101  1 !               FDL$RELEASE in FDL$CREATE - to let
;   103          0102  1 !               regression tests of CJF work.
;   104          0103  1 !
;   105          0104  1 !       V03-006 KFH0006         Ken Henderson           6 Jan 1983
;   106          0105  1 !               Fixed allocation of NAM block in
;   107          0106  1 !               FDL$CREATE
;   108          0107  1 !
;   109          0108  1 !       V03-005 KFH0005         Ken Henderson           5 Jan 1983
;   110          0109  1 !               Moved alloc/dealloc of FDL$AB_AREA_BKZ
;   111          0110  1 !               to FDL$$CHECK_BLOCKS in FDLGEN.B32
;   112          0111  1 !
;   113          0112  1 !       V03-004 KFH0004         Ken Henderson           4 Jan 1983
;   114          0113  1 !               Added allocation and deallocation
;   115          0114  1 !               of FDL$AB_AREA_BKZ in FDL$RELEASE
;   116          0115  1 !
;   117          0116  1 !       V03-003 KFH0003         Ken Henderson           22 Nov 1982
;   118          0117  1 !               Mods to FDL$PARSE to support
;   119          0118  1 !               fdl-dflt-spc and stmnt-num
;   120          0119  1 !               Mods to FDL$CREATE to support
;   121          0120  1 !               stmnt-num
;   122          0121  1 !               Fixed call to LIB$SIG_TO_RET
;   123          0122  1 !               Finished FDL$RELEASE, FDL$GENERATE
;   124          0123  1 !
;   125          0124  1 !       V03-002 KFH0002         Ken Henderson           6 Oct 1982
;   126          0125  1 !               Added FDL$AB_PARSED_RAB
;   127          0126  1 !               Changed FDL$$PARSE to FDL$PARSE
;   128          0127  1 !               Added FDL$GENERATE, FDL$RELEASE
;   129          0128  1 !
;   130          0129  1 !       V03-001 KFH0001         Ken Henderson           26 March 1982
;   131          0130  1 !               Added calls to LIB$ANALYZE_SDESC to generalize string
;   132          0131  1 !               handling; and added call to LIB$SCOPY_DXDX to move
;   133          0132  1 !               the result string to the output also add buffer for
;   134          0133  1 !               upcasing to.
;   135          0134  1 !****
```

```
;  137     0135  1
;  138     0136  1 PSECT
;  139     0137  1         OWN     = _FDL$OWN      (PIC),
;  140     0138  1         GLOBAL  = _FDL$GLOBAL   (PIC),
;  141     0139  1         PLIT    = _FDL$PLIT     (SHARE,PIC),
;  142     0140  1         CODE    = _FDL$CODE     (SHARE,PIC);
;  143     0141  1
;  144     0142  1 LIBRARY 'SYS$LIBRARY:STARLET';
;  145     0143  1 REQUIRE 'SRC$:FDLUTIL';
;  146     0328  1 REQUIRE 'LIB$:FDLPARDEF';
;  147     0867  1
;  148     0868  1 EXTERNAL ROUTINE
;  149     0869  1         LIB$ANALYZE_SDESC,
;  150     0870  1         LIB$SCOPY_DXDX,
;  151     0871  1         LIB$SCOPY_R_DX,
;  152     0872  1         LIB$ESTABLISH,
;  153     0873  1         LIB$SIG_TO_RET,
;  154     0874  1         LIB$GET_VM,
;  155     0875  1         LIB$FREE_VM,
;  156     0876  1         LIB$TPARSE,
;  157     0877  1         FDL$$CHECK_BLOCKS,
;  158     0878  1         FDL$$GEN_SPEC,
;  159     0879  1         FDL$$GET_VM,
;  160     0880  1         FDL$$FREE_VM,
;  161     0881  1         FDL$$INIT_PARSE         : NOVALUE,
;  162     0882  1         FDL$$FINISH_PARSE,
;  163     0883  1         FDL$$RMS_OPEN_ERROR     : NOVALUE;
;  164     0884  1
;  165     0885  1 FORWARD ROUTINE
;  166     0886  1         FDL$PARSE,
;  167     0887  1         FDL$RELEASE,                                            ;
;  168     0888  1         HANDLER;
;  169     0889  1
;  170     0890  1 EXTERNAL                                                        ;
;  171     0891  1         FDL$AB_CTRL             : BLOCK [ ,BYTE ],
;  172     0892  1         FDL$AB_BLOCK_BLK        : VECTOR [ 4,LONG ],
;  173     0893  1         FDL$GL_PCALL,
;  174     0894  1         FDL$GL_INVBLK_PTR,
;  175     0895  1         FDL$GL_STNUMPTR,
;  176     0896  1         FDL$GL_MAXLINE,
;  177     0897  1         FDL$AB_OUT_STRING       : REF DESC_BLK,
;  178     0898  1         FDL$AB_FDL_STRING       : DESC_BLK,
;  179     0899  1         FDL$AB_LINE             : DESC_BLK,
;  180     0900  1         FDL$AB_UPCASED          : DESC_BLK,
;  181     0901  1         FDL$AB_KEY_TABLE,
;  182     0902  1         FDL$AB_STATE_TABLE,
;  183     0903  1         FDL$AB_TPARSE_BLOCK     : BLOCK [ ,BYTE ];
;  184     0904  1
;  185     0905  1 DEFINE_ERROR_CODES;
;  186     0906  1
;  187     0907  1 GLOBAL
;  188     0908  1         FDL$AB_FDL_RAB          : $RAB_DECL,
;  189     0909  1         FDL$AB_PARSED_FAB       : REF BLOCK [ ,BYTE ],
;  190     0910  1         FDL$AB_PARSED_RAB       : REF BLOCK [ ,BYTE ];
;  191     0911  1
;  192     0912  1 GLOBAL BIND
;  193     0913  1         FDL$AB_GENFAB = FDL$AB_PARSED_FAB : REF BLOCK [ ,BYTE ],
```

```
;  194          0914  1          FDL$AB_GENRAB = FDL$AB_PARSED_RAB : REF BLOCK [ ,BYTE ];
```

```
;  196     0915  1 %SBTTL  'FDL$CREATE'
;  197     0916  1 GLOBAL ROUTINE FDL$CREATE  =
;  198     0917  1 !++
;  199     0918  1 !
;  200     0919  1 ! Functional Description:
;  201     0920  1 !
;  202     0921  1 !       This routine calls FDL$PARSE to parse the FDL spec and then
;  203     0922  1 !       creates the file and closes it.  It is called by CREATE/FDL.
;  204     0923  1 !
;  205     0924  1 ! Calling Sequence:
;  206     0925  1 !
;  207     0926  1 !       fdl$create(     fdl_desc
;  208     0927  1 !                       [,file_name]
;  209     0928  1 !                       [,default_name]
;  210     0929  1 !                       [,result_name]
;  211     0930  1 !                       [,fid_block]
;  212     0931  1 !                       [,flags]
;  213     0932  1 !                       [,stmnt-num]
;  214     0933  1 !                       [,retlen]
;  215     0934  1 !                       [,sts]
;  216     0935  1 !                       [,stv] )
;  217     0936  1 !
;  218     0937  1 ! Input Parameters:
;  219     0938  1 !
;  220     0939  1 !       fdl_desc        - descriptor of the fdl file name string (required)
;  221     0940  1 !       file_name       - descriptor file name to overide the name specified
;  222     0941  1 !                         in the fdl file (optional)
;  223     0942  1 !       default_name    - descriptor default file name to overide the default
;  224     0943  1 !                         name specified in the fdl file (optional)
;  225     0944  1 !
;  226     0945  1 !       flags           - address of flags longword (optional)
;  227     0946  1 !                 FDL$V_SIGNAL            signal errors instead of returning
;  228     0947  1 !                 FDL$V_FDL_STRING        input fdl-spec is a char string
;  229     0948  1 !                 FDL$V_$CALLBACK         used by EDF
;  230     0949  1 !
;  231     0950  1 ! Implicit Inputs:
;  232     0951  1 !       none
;  233     0952  1 !
;  234     0953  1 ! Output Parameters:
;  235     0954  1 !
;  236     0955  1 !       result_name     - descriptor to receive the file name which was created
;  237     0956  1 !                         (optional)
;  238     0957  1 !
;  239     0958  1 !       fid_block       - address of a 3 longword used block to receive the fid
;  240     0959  1 !                         of the file created (optional)
;  241     0960  1 !
;  242     0961  1 !       stmnt-num       - address of longword to recieve statement
;  243     0962  1 !                         number (optional)
;  244     0963  1 !
;  245     0964  1 !       retlen          - address of longword to recieve length of resultname
;  246     0965  1 !                         string (optional)
;  247     0966  1 !
;  248     0967  1 !       sts             - address of longword to recieve FAB$L_STS from $create
;  249     0968  1 !
;  250     0969  1 !       stv             - address of longword to recieve FAB$L_STV from $create
;  251     0970  1 !
;  252     0971  1 ! Implicit Outputs:
```

```
;  253     0972  1 !       none
;  254     0973  1 !
;  255     0974  1 ! Routine Value:
;  256     0975  1 !
;  257     0976  1 !       success or error code
;  258     0977  1 !
;  259     0978  1 ! Side Effects:
;  260     0979  1 !       none
;  261     0980  1 !
;  262     0981  1 !--
;  263     0982  1
;  264     0983  2     BEGIN
;  265     0984  2
;  266     0985  2     BUILTIN
;  267     0986  2         ACTUALCOUNT,
;  268     0987  2         ACTUALPARAMETER;
;  269     0988  2
;  270     0989  2     OWN
;  271     0990  2         CREATE_DESC     : DESC_BLK;
;  272     0991  2
;  273     0992  2     LOCAL
;  274     0993  2         ESA_ADDR,
;  275     0994  2         RSA_ADDR,
;  276     0995  2         COUNT,
;  277     0996  2         FLAGS           : REF BLOCK [ ,BYTE ],
;  278     0997  2         CREATE_FAB      : REF BLOCK [ ,BYTE ],
;  279     0998  2         CREATE_RAB      : REF BLOCK [ ,BYTE ],
;  280     0999  2         NAME_BLOCK      : REF BLOCK [ ,BYTE ],
;  281     1000  2         XSTATUS         : LONG,
;  282     1001  2         STATUS          : LONG;
;  283     1002  2
;  284     1003  2     BIND
;  285     1004  2         STATUS_CODE     = STATUS : BLOCK [ 4,BYTE ];
;  286     1005  2
;  287     1006  2     ! Set up handler to control errors
;  288     1007  2     !
;  289     1008  2     LIB$ESTABLISH ( HANDLER );
;  290     1009  2
;  291     1010  2     ! Clear the valid stmnt flag as a fix...  should always be cleared on entry
;  292     1011  2     !
;  293     1012  2     FDL$AB_CTRL [ FDL$V_STVALID ] = _CLEAR;
;  294     1013  2
;  295     1014  2     ! Check the validity of the call
;  296     1015  2     !
;  297     1016  2     ! Get the number of arguments
;  298     1017  2     !
;  299     1018  2     COUNT = ACTUALCOUNT();
;  300     1019  2
;  301     1020  2     ! Check if the number is legal ie. 1 thru 10
;  302     1021  2     !
;  303     1022  3     IF ( .COUNT LSS 1 ) OR ( .COUNT GTR 10 )
;  304     1023  2     THEN
;  305     1024  2         RETURN FDL$_ILL_ARG;
;  306     1025  2
;  307     1026  2     ! Get flags argument - to pass it on
;  308     1027  2     !
;  309     1028  3     IF ( .COUNT GTR 5 )
```

```
;  310     1029  2      THEN
;  311     1030  2          FLAGS = ACTUALPARAMETER (6)
;  312     1031  2      ELSE
;  313     1032  2          FLAGS = 0;
;  314     1033  2
;  315     1034  2      ! Get stmnt-num argument
;  316     1035  2      !
;  317     1036  2      IF .COUNT GTR 6
;  318     1037  2      THEN
;  319     1038  2          FDL$GL_STNUMPTR = ACTUALPARAMETER (7)
;  320     1039  2      ELSE
;  321     1040  2          FDL$GL_STNUMPTR = 0;
;  322     1041  2
;  323     1042  2      IF .FDL$GL_STNUMPTR NEQU 0
;  324     1043  2      THEN
;  325     1044  2          FDL$AB_CTRL [ FDL$V_STVALID ] = _SET
;  326     1045  2      ELSE
;  327     1046  2          FDL$AB_CTRL [ FDL$V_STVALID ] = _CLEAR;
;  328     1047  2
;  329     1048  3      BEGIN
;  330     1049  3
;  331     1050  3      LOCAL
;  332     1051  3          FDL_DESC_PTR;
;  333     1052  3
;  334     1053  3
;  335     1054  3      ! The first argument should be the FDL file spec.
;  336     1055  3      !
;  337     1056  3      FDL_DESC_PTR = ACTUALPARAMETER(1);
;  338     1057  3
;  339     1058  3      ! Parse the FDL file
;  340     1059  3      !
;  341     1060  4      IF NOT ( STATUS = FDL$PARSE( .FDL_DESC_PTR,CREATE_FAB,CREATE_RAB,.FLAGS,0,
;  342     1061  4                                   .FDL$GL_STNUMPTR ) )
;  343     1062  3      THEN
;  344     1063  3
;  345     1064  3          ! If there was a warning just signal it and try to create the file
;  346     1065  3          ! else return (error and severe)
;  347     1066  3          !
;  348     1067  4          IF ( .STATUS_CODE [ STS$V_SEVERITY ] EQLU STS$K_WARNING )
;  349     1068  3          THEN
;  350     1069  3              SIGNAL( .STATUS )
;  351     1070  3          ELSE
;  352     1071  3              RETURN .STATUS
;  353     1072  3
;  354     1073  2      END;
;  355     1074  2
;  356     1075  2
;  357     1076  2      ! See it the a second parameter was given. If so the this is the name
;  358     1077  2      ! to be used for the created file.
;  359     1078  2      !
;  360     1079  3      IF ( .COUNT GTR 1 )
;  361     1080  2      THEN
;  362     1081  3          BEGIN
;  363     1082  3
;  364     1083  3          LOCAL   NAME_DESC : REF DESC_BLK;
;  365     1084  3
;  366     1085  3          NAME_DESC = ACTUALPARAMETER(2);
```

```
;  367     1086  3
;  368     1087  4          IF ( .NAME_DESC NEQ 0 )
;  369     1088  3          THEN
;  370     1089  4              BEGIN
;  371     1090  4
;  372     1091  4              LOCAL
;  373     1092  4                      LENGTH      : WORD,
;  374     1093  4                      ADDR        : LONG;
;  375     1094  4
;  376     1095  4              ! Allow for wierd strings (like VARYING or byte-arrays)
;  377     1096  4              !
;  378     1097  4              RET_ON_ERROR( LIB$ANALYZE_SDESC( .NAME_DESC,LENGTH,ADDR ) );
;  379     1098  4
;  380     1099  5              IF ( .LENGTH NEQ 0 ) AND ( .ADDR NEQ 0 )
;  381     1100  4              THEN
;  382     1101  5                  BEGIN
;  383     1102  5
;  384     1103  5                  ! If a file name was given to replace any given in the FDL spec.
;  385     1104  5                  !
;  386     1105  5                  CREATE_FAB [ FAB$B_FNS ] = .LENGTH;
;  387     1106  5                  CREATE_FAB [ FAB$L_FNA ] = .ADDR;
;  388     1107  5
;  389     1108  4                  END;
;  390     1109  3              END;
;  391     1110  2          END;
;  392     1111  2
;  393     1112  2      ! See if the 3rd parameter was given. If so this is the default name
;  394     1113  2      ! to be used for the created file.
;  395     1114  2      !
;  396     1115  3      IF ( .COUNT GTR 2 )
;  397     1116  2      THEN
;  398     1117  3          BEGIN
;  399     1118  3
;  400     1119  3          LOCAL
;  401     1120  3              DEFAULT_DESC : REF DESC_BLK;
;  402     1121  3
;  403     1122  3          DEFAULT_DESC = ACTUALPARAMETER(3);
;  404     1123  3
;  405     1124  4          IF ( .DEFAULT_DESC NEQ 0 )
;  406     1125  3          THEN
;  407     1126  4              BEGIN
;  408     1127  4
;  409     1128  4              LOCAL
;  410     1129  4                      LENGTH      : WORD,
;  411     1130  4                      ADDR        : LONG;
;  412     1131  4
;  413     1132  4              ! Allow for wierd strings (like VARYING or byte-arrays)
;  414     1133  4              !
;  415     1134  4              RET_ON_ERROR( LIB$ANALYZE_SDESC( .DEFAULT_DESC,LENGTH,ADDR ) );
;  416     1135  4
;  417     1136  5              IF ( .LENGTH NEQ 0 ) AND ( .ADDR NEQ 0 )
;  418     1137  4              THEN
;  419     1138  5                  BEGIN
;  420     1139  5
;  421     1140  5                  ! If a default name was given, put it in the FAB as well.
;  422     1141  5                  !
;  423     1142  5                  CREATE_FAB [ FAB$B_DNS ] = .LENGTH;
```

```
;  424          1143  5                  CREATE_FAB [ FAB$L_DNA ] = .ADDR;
;  425          1144  5
;  426          1145  4                  END;
;  427          1146  3              END;
;  428          1147  2          END;
;  429          1148  2
;  430          1149  2      ! Build a name block for the fab returned by fdl$$parse
;  431          1150  2      !
;  432          1151  2      ! Allocate the space for the name block and buffers:
;  433          1152  2      !
;  434          1153  2      !                 ------------------------
;  435          1154  2      !                 |         nam          |  nam$k_bln
;  436          1155  2      !                 ------------------------
;  437          1156  2      !                 |   expanded str buffer|  esa_buf_siz
;  438          1157  2      !                 ------------------------
;  439          1158  2      !                 |  resultant str buffer|  rsa_buf_siz
;  440          1159  2      !                 ------------------------
;  441          1160  2      !
;  442          1161  2      !
;  443          1162  2      NAME_BLOCK = FDL$$GET_VM( NAM$K_BLN );
;  444          1163  2      ESA_ADDR = FDL$$GET_VM( ESA_BUF_SIZ );
;  445          1164  2      RSA_ADDR = FDL$$GET_VM( RSA_BUF_SIZ );
;  446          1165  2
;  447          1166  2      ! Init the name block
;  448          1167  2      !
;  449        P 1168  2      $NAM_INIT ( NAM = .NAME_BLOCK,
;  450        P 1169  2                  ESA = .ESA_ADDR,
;  451        P 1170  2                  ESS = ESA_BUF_SIZ,
;  452        P 1171  2                  RSA = .RSA_ADDR,
;  453          1172  2                  RSS = RSA_BUF_SIZ );
;  454          1173  2
;  455          1174  2      ! Connect the name block
;  456          1175  2      !
;  457          1176  2      CREATE_FAB [ FAB$L_NAM ] = .NAME_BLOCK;
;  458          1177  2
;  459          1178  2      ! Create the file
;  460          1179  2      !
;  461          1180  2      ! Errors will be create error
;  462          1181  2      !
;  463          1182  2      CREATE_FAB [ FAB$L_CTX ] = FDL$_CREATE;
;  464          1183  2
;  465          1184  2      ! If this is from dcl then set the error handler to signal fdl error
;  466          1185  2      ! and the rms error else return normally
;  467          1186  2      !
;  468          1187  2      IF .FDL$AB_CTRL [ FDL$V_DCL ]
;  469          1188  2      THEN
;  470          1189  3          STATUS = $CREATE( FAB=.CREATE_FAB,ERR=FDL$$RMS_OPEN_ERROR )
;  471          1190  2      ELSE
;  472          1191  2          STATUS = $CREATE( FAB=.CREATE_FAB );
;  473          1192  2
;  474          1193  2      ! Output the STS if it's required.
;  475          1194  2      !
;  476          1195  2      IF .COUNT GTR 8
;  477          1196  2      THEN
;  478          1197  3          BEGIN
;  479          1198  3
;  480          1199  3          LOCAL
```

```
;  481     1200  3              STS         : LONG;
;  482     1201  3
;  483     1202  3          STS     = ACTUALPARAMETER (9);
;  484     1203  3
;  485     1204  3          IF .STS NEQ 0
;  486     1205  3          THEN
;  487     1206  3
;  488     1207  3              .STS = .CREATE_FAB [ FAB$L_STS ];
;  489     1208  3
;  490     1209  2          END;
;  491     1210  2
;  492     1211  2      ! Output the STV if it's required.
;  493     1212  2      !
;  494     1213  2      IF .COUNT GTR 9
;  495     1214  2      THEN
;  496     1215  3          BEGIN
;  497     1216  3
;  498     1217  3          LOCAL
;  499     1218  3              STV         : LONG;
;  500     1219  3
;  501     1220  3          STV     = ACTUALPARAMETER (10);
;  502     1221  3
;  503     1222  3          IF .STV NEQ 0
;  504     1223  3          THEN
;  505     1224  3
;  506     1225  3              .STV = .CREATE_FAB [ FAB$L_STV ];
;  507     1226  3
;  508     1227  2          END;
;  509     1228  2
;  510     1229  2      ! If there was an error creating the file return fdl$_create to the user
;  511     1230  2      !
;  512     1231  2      IF NOT .STATUS
;  513     1232  2      THEN
;  514     1233  2          STATUS = FDL$_CREATE;
;  515     1234  2
;  516     1235  2      $CLOSE( FAB=.CREATE_FAB );
;  517     1236  2
;  518     1237  2      ! See if the 4th parameter was given. If so return the result name.
;  519     1238  2      ! NOTE: This is where the user can get access to allocated memory
;  520     1239  2      !
;  521     1240  3      IF ( .COUNT GTR 3 )
;  522     1241  2      THEN
;  523     1242  3          BEGIN
;  524     1243  3
;  525     1244  3          LOCAL
;  526     1245  3              RETLEN      : LONG,
;  527     1246  3              RESULT_DESC : REF DESC_BLK,
;  528     1247  3              TEMP_DESC   : DESC_BLK PRESET ( [ DSC$B_CLASS ] = DSC$K_CLASS_S,
;  529     1248  3                                              [ DSC$B_DTYPE ] = DSC$K_DTYPE_T );
;  530     1249  3
;  531     1250  3          RESULT_DESC = ACTUALPARAMETER(4);
;  532     1251  3
;  533     1252  3          IF .RESULT_DESC NEQ 0
;  534     1253  3          THEN
;  535     1254  4              BEGIN
;  536     1255  4
;  537     1256  4              IF ..RESULT_DESC NEQ 0
```

```
;  538     1257  4              THEN
;  539     1258  5                  BEGIN
;  540     1259  5
;  541     1260  5                  ! If a result name desc was given, stuff it with
;  542     1261  5                  ! the best name for file which was created (or attempted)
;  543     1262  5                  !
;  544     1263  5                  ! First try the resultant string
;  545     1264  5                  !
;  546     1265  5                  IF .NAME_BLOCK [ NAM$B_RSL ] NEQU 0
;  547     1266  5                  THEN
;  548     1267  6                      BEGIN
;  549     1268  6
;  550     1269  6                      TEMP_DESC [ DSC$W_LENGTH ] = .NAME_BLOCK [ NAM$B_RSL ];
;  551     1270  6                      TEMP_DESC [ DSC$A_POINTER ] = .NAME_BLOCK [ NAM$[_RSA ];
;  552     1271  6
;  553     1272  7                      RET_ON_ERROR( LIB$SCOPY_DXDX( TEMP_DESC,.RESULT_DESC ) )
;  554     1273  7
;  555     1274  6                      END
;  556     1275  6
;  557     1276  6                  ! Next try the expanded string
;  558     1277  6                  !
;  559     1278  5                  ELSE IF .NAME_BLOCK [ NAM$B_ESL ] NEQU 0
;  560     1279  5                  THEN
;  561     1280  6                      BEGIN
;  562     1281  6
;  563     1282  6                      TEMP_DESC [ DSC$W_LENGTH ] = .NAME_BLOCK [ NAM$B_ESL ];
;  564     1283  6                      TEMP_DESC [ DSC$A_POINTER ] = .NAME_BLOCK [ NAM$[_ESA ];
;  565     1284  6
;  566     1285  7                      RET_ON_ERROR( LIB$SCOPY_DXDX( TEMP_DESC,.RESULT_DESC ) )
;  567     1286  7
;  568     1287  6                      END
;  569     1288  6
;  570     1289  6                  ! If all else fails use the name string
;  571     1290  6                  !
;  572     1291  5                  ELSE
;  573     1292  6                      BEGIN
;  574     1293  6
;  575     1294  6                      TEMP_DESC [ DSC$W_LENGTH ] = .CREATE_FAB [ FAB$B_FNS ];
;  576     1295  6                      TEMP_DESC [ DSC$A_POINTER ] = .CREATE_FAB [ FAB$[_FNA ];
;  577     1296  6
;  578     1297  7                      RET_ON_ERROR( LIB$SCOPY_DXDX( TEMP_DESC,.RESULT_DESC ) )
;  579     1298  7
;  580     1299  5                      END;
;  581     1300  5
;  582     1301  5                  ! Output the length of the resultname string if required.
;  583     1302  5                  !
;  584     1303  5                  IF .COUNT GTR 7
;  585     1304  5                  THEN
;  586     1305  6                      BEGIN
;  587     1306  6
;  588     1307  6                      RETLEN      = ACTUALPARAMETER (8);
;  589     1308  6
;  590     1309  6                      IF .RETLEN NEQ 0
;  591     1310  6                      THEN
;  592     1311  7                          BEGIN
;  593     1312  7
;  594     1313  7                          .RETLEN = .TEMP_DESC [ DSC$W_LENGTH ];
```

```
;  595          1314  7
;  596          1315  6                          END;
;  597          1316  5                      END;
;  598          1317  4                  END;
;  599          1318  3              END;
;  600          1319  2          END;
;  601          1320  2
;  602          1321  2      ! If the caller wants a file id get it
;  603          1322  2      !
;  604          1323  3      IF ( .COUNT GTR 4 )
;  605          1324  2      THEN
;  606          1325  3          BEGIN
;  607          1326  3
;  608          1327  3          LOCAL FID_BLOCK : REF VECTOR [,LONG ];
;  609          1328  3
;  610          1329  3          FID_BLOCK = ACTUALPARAMETER(5);
;  611          1330  3
;  612          1331  3          IF .FID_BLOCK NEQ 0
;  613          1332  3          THEN
;  614          1333  4              BEGIN
;  615          1334  4
;  616          1335  4              FID_BLOCK [0] = .NAME_BLOCK [ NAM$W_FID_NUM ];
;  617          1336  4              FID_BLOCK [1] = .NAME_BLOCK [ NAM$W_FID_SEQ ];
;  618          1337  4              FID_BLOCK [2] = .NAME_BLOCK [ NAM$W_FID_RVN ];
;  619          1338  4
;  620          1339  3              END;
;  621          1340  2          END;
;  622          1341  2
;  623          1342  2      ! Deallocate the memory used during the parse
;  624          1343  2      !
;  625          1344  2  !*****************************************************************************
;  626          1345  2  !
;  627          1346  2  ! THIS CALL TO FDL$RELEASE IS THE MOST TROUBLESOME PART OF FDL$CREATE!!!
;  628          1347  2  ! FDL$RELEASE CONTAINS AN OBSCURE ERROR IN USING LIB$FREE_VM AND IT
;  629          1348  2  ! MUST BE FIXED FOR FT2.  UNTIL THEN, FDL$CREATE WILL WORK FINE BY JUST
;  630          1349  2  ! NOT CALLING FDL$RELEASE - AND FDL$CREATE GETS VERY HEAVY USE.
;  631          1350  2  !
;  632          1351  2  !*****************************************************************************
;  633          1352  2  !    RET_ON_ERROR ( FDL$RELEASE( CREATE_FAB,CREATE_RAB,,.FLAGS ) );
;  634          1353  2  !*****************************************************************************
;  635          1354  2
;  636          1355  2      RETURN .STATUS;
;  637          1356  2
;  638          1357  1      END;


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

                                .PSECT  _FDL$GLOBAL,NOEXE,  PIC,2

                    00000 FDL$AB_FDL_RAB::
                                .BLKB   68
                    00044 FDL$AB_PARSED_FAB::
                                .BLKB   4
                    00048 FDL$AB_PARSED_RAB::
                                .BLKB   4
```

```
                        .PSECT  _FDL$OWN,NOEXE,  PIC,2

00000 CREATE_DESC:
                        .BLKB   8

      FDL$AB_GENFAB==       FDL$AB_PARSED_FAB
      FDL$AB_GENRAB==       FDL$AB_PARSED_RAB
                        .EXTRN  LIB$ANALYZE_SDESC
                        .EXTRN  LIB$SCOPY_DXDX, LIB$SCOPY_R_DX
                        .EXTRN  LIB$ESTABLISH, LIB$SIG_TO_RET
                        .EXTRN  LIB$GET_VM, LIB$FREE_VM
                        .EXTRN  LIB$TPARSE, FDL$$CHECK_BLOCKS
                        .EXTRN  FDL$$GEN_SPEC, FDL$$GET_VM
                        .EXTRN  FDL$$FREE_VM, FDL$$INIT_PARSE
                        .EXTRN  FDL$$FINISH_PARSE
                        .EXTRN  FDL$$RMS_OPEN_ERROR
                        .EXTRN  FDL$AB_CTRL, FDL$AB_BLOCK_BLK
                        .EXTRN  FDL$GL_PCALL, FDL$GL_INVBLK_PTR
                        .EXTRN  FDL$GL_STNUMPTR
                        .EXTRN  FDL$GL_MAXLINE, FDL$AB_OUT_STRING
                        .EXTRN  FDL$AB_FDL_STRING
                        .EXTRN  FDL$AB_LINE, FDL$AB_UPCASED
                        .EXTRN  FDL$AB_KEY_TABLE
                        .EXTRN  FDL$AB_STATE_TABLE
                        .EXTRN  FDL$AB_TPARSE_BLOCK
                        .EXTRN  FDL$_FACILITY, FDL$_FAO_MAX
                        .EXTRN  FDL$_ABKW, FDL$_ABPRIKW
                        .EXTRN  FDL$_CREATE, FDL$_CREATED
                        .EXTRN  FDL$_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  SYS$CREATE, SYS$CLOSE

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

                    0FFC 00000          .ENTRY  FDL$CREATE, Save R2,R3,R4,R5,R6,R7,R8,R9,-      ; 0916
                                                R10,R11                                         ;
            5B 00000000G   00   9E 00002        MOVAB   FDL$AB_CTRL, R11                        ;
            5E             20   C2 00009        SUBL2   #32, SP                                 ;
               00000000V   00   9F 0000C        PUSHAB  HANDLER                                 ; 1008
00000000G   00             01   FB 00012        CALLS   #1, LIB$ESTABLISH                       ;
      02    AB             04   8A 00019        BICB2   #4, FDL$AB_CTRL+2                       ; 1012
            57             6C   9A 0001D        MOVZBL  (AP), COUNT                             ; 1018
                           05   15 00020        BLEQ    1$                                      ; 1022
            0A             57   D1 00022        CMPL    COUNT, #10                              ;
                           08   15 00025        BLEQ    2$                                      ;
            50 00000000G   8F   D0 00027 1$:    MOVL    #FDL$_ILL_ARG, R0                       ; 1024
```

```
                                        04 0002E        RET
                                05          57  D1 0002F 2$:    CMPL    COUNT, #5                               ;1028
                                            06  15 00032        BLEQ    3$
                                51      18  AC  D0 00034        MOVL    24(AP), FLAGS                           ;1030
                                            02  11 00038        BRB     4$
                                            51  D4 0003A 3$:    CLRL    FLAGS                                   ;1032
                                06          57  D1 0003C 4$:    CMPL    COUNT, #6                               ;1036
                                            0A  15 0003F        BLEQ    5$
                  00000000G     00      1C  AC  D0 00041        MOVL    28(AP), FDL$GL_STNUMPTR                 ;1038
                                            06  11 00049        BRB     6$
                            00000000G   00  D4 0004B 5$:    CLRL    FDL$GL_STNUMPTR                         ;1040
                                50 00000000G   00  D0 00051 6$:    MOVL    FDL$GL_STNUMPTR, R0                     ;1042
                                            06  13 00058        BEQL    7$
                        02      AB          04  88 0005A        BISB2   #4, FDL$AB_CTRL+2                       ;1044
                                            04  11 0005E        BRB     8$
                        02      AB          04  8A 00060 7$:    BICB2   #4, FDL$AB_CTRL+2                       ;1046
                                52      04  AC  D0 00064 8$:    MOVL    4(AP), FDL_DESC_PTR                     ;1056
                                            50  DD 00068        PUSHL   R0                                      ;1061
                                            7E  D4 0006A        CLRL    -(SP)                                   ;1060
                                            51  DD 0006C        PUSHL   FLAGS
                                        0C  AE  9F 0006E        PUSHAB  CREATE_RAB
                                        14  AE  9F 00071        PUSHAB  CREATE_FAB
                                            52  DD 00074        PUSHL   FDL_DESC_PTR
                  00000000V     00          06  FB 00076        CALLS   #6, FDL$PARSE
                                58          50  D0 0007D        MOVL    R0, STATUS
                                11          58  E8 00080        BLBS    STATUS, 10$
                                07          58  93 00083        BITB    STATUS_CODE, #7                         ;1067
                                            03  13 00086        BEQL    9$
                                        019B    31 00088        BRW     24$
                                            58  DD 0008B 9$:    PUSHL   STATUS                                  ;1069
                  00000000G     00          01  FB 0008D        CALLS   #1, LIB$SIGNAL
                                01          57  D1 00094 10$:   CMPL    COUNT, #1                               ;1079
                                            30  15 00097        BLEQ    11$
                                50      08  AC  D0 00099        MOVL    8(AP), NAME_DESC                        ;1085
                                            2A  13 0009D        BEQL    11$                                     ;1087
                                        08  AE  9F 0009F        PUSHAB  ADDR                                    ;1097
                                        10  AE  9F 000A2        PUSHAB  LENGTH
                                            50  DD 000A5        PUSHL   NAME_DESC
                  00000000G     00          03  FB 000A7        CALLS   #3, LIB$ANALYZE_SDESC
                                32          50  E9 000AE        BLBC    STATUS, 12$
                                        0C  AE  B5 000B1        TSTW    LENGTH                                  ;1099
                                            13  13 000B4        BEQL    11$
                                        08  AE  D5 000B6        TSTL    ADDR
                                            0E  13 000B9        BEQL    11$
                                50      04  AE  D0 000BB        MOVL    CREATE_FAB, R0                          ;1105
                        34      A0      0C  AE  90 000BF        MOVB    LENGTH, 52(R0)
                        2C      A0      08  AE  D0 000C4        MOVL    ADDR, 44(R0)                            ;1107
                                02          57  D1 000C9 11$:   CMPL    COUNT, #2                               ;1115
                                            31  15 000CC        BLEQ    14$
                                50      0C  AC  D0 000CE        MOVL    12(AP), DEFAULT_DESC                    ;1122
                                            2B  13 000D2        BEQL    14$                                     ;1124
                                        10  AE  9F 000D4        PUSHAB  ADDR                                    ;1134
                                        18  AE  9F 000D7        PUSHAB  LENGTH
                                            50  DD 000DA        PUSHL   DEFAULT_DESC
                  00000000G     00          03  FB 000DC        CALLS   #3, LIB$ANALYZE_SDESC
                                01          50  E8 000E3 12$:   BLBS    STATUS, 13$
                                            04  000E6        RET
```

```
                                                14   AE  B5 000E7 13$:    TSTW    LENGTH                                  ; 1136
                                                13   13 000EA          BEQL    14$                                     ;
                                                10   AE  D5 000EC          TSTL    ADDR                                    ;
                                                0E   13 000EF          BEQL    14$                                     ;
                                           50   04   AE  D0 000F1          MOVL    CREATE_FAB, R0                          ; 1142
                                        35 A0   14   AE  90 000F5          MOVB    LENGTH, 53(R0)                          ;
                                        30 A0   10   AE  D0 000FA          MOVL    ADDR, 48(R0)                            ; 1143
                                        7E      60   8F  9A 000FF 14$:     MOVZBL  #96, -(SP)                              ; 1162
                                00000000G 00         01  FB 00103          CALLS   #1, FDL$$GET_VM                         ;
                                        56           50  D0 0010A          MOVL    R0, NAME_BLOCK                          ;
                                        7E      FF   8F  9A 0010D          MOVZBL  #255, -(SP)                             ; 1163
                                00000000G 00         01  FB 00111          CALLS   #1, FDL$$GET_VM                         ;
                                        5A           50  D0 00118          MOVL    R0, ESA_ADDR                            ;
                                        7E      FF   8F  9A 0011B          MOVZBL  #255, -(SP)                             ; 1164
                                00000000G 00         01  FB 0011F          CALLS   #1, FDL$$GET_VM                         ;
                                        59           50  D0 00126          MOVL    R0, RSA_ADDR                            ;
0060  8F                  00            6E           00  2C 00129          MOVC5   #0, (SP), #0, #96, (NAME_BLOCK)         ; 1172
                                                     66  00130                                                           ;
                                        66      6002 8F  B0 00131          MOVW    #24578, (NAME_BLOCK)                    ;
                                     02 A6           01  8E 00136          MNEGB   #1, 2(NAME_BLOCK)                       ;
                                     04 A6           59  D0 0013A          MOVL    RSA_ADDR, 4(NAME_BLOCK)                 ;
                                     0A A6           01  8E 0013E          MNEGB   #1, 10(NAME_BLOCK)                      ;
                                     0C A6           5A  D0 00142          MOVL    ESA_ADDR, 12(NAME_BLOCK)                ;
                                        52      04   AE  D0 00146          MOVL    CREATE_FAB, R2                          ; 1176
                                     28 A2           56  D0 0014A          MOVL    NAME_BLOCK, 40(R2)                      ;
                                     18 A2 00000000G 8F  D0 0014E          MOVL    #FDL$_CREATE, 24(R2)                    ; 1182
                                  11 01 AB           03  E1 00156          BBC     #3, FDL$AB_CTRL+1, 15$                  ; 1187
                                           00000000G 00  9F 0015B          PUSHAB  FDL$$RMS_OPEN_ERROR                     ; 1189
                                                     52  DD 00161          PUSHL   R2                                      ;
                                00000000G 00         02  FB 00163          CALLS   #2, SYS$CREATE                          ;
                                                     09  11 0016A          BRB     16$                                     ;
                                                     52  DD 0016C 15$:     PUSHL   R2                                      ; 1191
                                00000000G 00         01  FB 0016E          CALLS   #1, SYS$CREATE                          ;
                                        58           50  D0 00175 16$:     MOVL    R0, STATUS                              ;
                                        08           57  D1 00178          CMPL    COUNT, #8                               ; 1195
                                                     0A  15 0017B          BLEQ    17$                                     ;
                                        50      24   AC  D0 0017D          MOVL    36(AP), STS                             ; 1202
                                                     04  13 00181          BEQL    17$                                     ; 1204
                                        60      08   A2  D0 00183          MOVL    8(R2), (STS)                            ; 1207
                                        09           57  D1 00187 17$:     CMPL    COUNT, #9                               ; 1213
                                                     0A  15 0018A          BLEQ    18$                                     ;
                                        50      28   AC  D0 0018C          MOVL    40(AP), STV                             ; 1220
                                                     04  13 00190          BEQL    18$                                     ; 1222
                                        60      0C   A2  D0 00192          MOVL    12(R2), (STV)                           ; 1225
                                        07           58  E8 00196 18$:     BLBS    STATUS, 19$                             ; 1231
                                        58 00000000G 8F  D0 00199          MOVL    #FDL$_CREATE, STATUS                    ; 1233
                                                     52  DD 001A0 19$:     PUSHL   R2                                      ; 1235
                                00000000G 00         01  FB 001A2          CALLS   #1, SYS$CLOSE                           ;
                                        03           57  D1 001A9          CMPL    COUNT, #3                               ; 1240
                                                     5F  15 001AC          BLEQ    23$                                     ;
                                  18 AE 010E0000     8F  D0 001AE          MOVL    #17694720, TEMP_DESC                    ; 1248
                                           1C        AE  D4 001B6          CLRL    TEMP_DESC+4                             ;
                                        53      10   AC  D0 001B9          MOVL    16(AP), RESULT_DESC                     ; 1250
                                                     4E  13 001BD          BEQL    23$                                     ; 1252
                                                     63  D5 001BF          TSTL    (RESULT_DESC)                           ; 1256
                                                     4A  13 001C1          BEQL    23$                                     ;
                                           03        A6  95 001C3          TSTB    3(NAME_BLOCK)                           ; 1265
```

```
                                 0C  13 001C6          BEQL    20$                                             ;
                         18  AE  03  A6 9B 001C8       MOVZBW  3(NAME_BLOCK), TEMP_DESC                        ; 1269
                         1C  AE  04  A6 D0 001CD       MOVL    4(NAME_BLOCK), TEMP_DESC+4                      ; 1270
                                 1B  11 001D2          BRB     22$                                             ; 1272
                             0B  A6  95 001D4 20$:     TSTB    11(NAME_BLOCK)                                  ; 1278
                                 0C  13 001D7          BEQL    21$                                             ;
                         18  AE  0B  A6 9B 001D9       MOVZBW  11(NAME_BLOCK), TEMP_DESC                       ; 1282
                         1C  AE  0C  A6 D0 001DE       MOVL    12(NAME_BLOCK), TEMP_DESC+4                     ; 1283
                                 0A  11 001E3          BRB     22$                                             ; 1285
                         18  AE  34  A2 9B 001E5 21$:  MOVZBW  52(R2), TEMP_DESC                               ; 1294
                         1C  AE  2C  A2 D0 001EA       MOVL    44(R2), TEMP_DESC+4                             ; 1295
                                 53  DD 001EF 22$:     PUSHL   RESULT_DESC                                     ; 1297
                                 1C  AE 9F 001F1       PUSHAB  TEMP_DESC                                       ;
                 00000000G       00      02 FB 001F4   CALLS   #2, CIB$SCOPY_DXDX                              ;
                                 2B      50 E9 001FB   BLBC    STATUS, 25$                                       ;
                                 07      57 D1 001FE   CMPL    COUNT, #7                                         ; 1303
                                         0A 15 00201   BLEQ    23$                                             ;
                                 50  20  AC  D0 00203   MOVL    32(AP), RETLEN                                  ; 1307
                                         04 13 00207   BEQL    23$                                             ; 1309
                                 60  18  AE  3C 00209   MOVZWL  TEMP_DESC, (RETLEN)                            ; 1313
                                 04      57 D1 0020D 23$:  CMPL    COUNT, #4                                     ; 1323
                                         14 15 00210   BLEQ    24$                                             ;
                                 50  14  AC  D0 00212   MOVL    20(AP), FID_BLOCK                               ; 1329
                                         0E 13 00216   BEQL    24$                                             ; 1331
                                 60  24  A6  3C 00218   MOVZWL  36(NAME_BLOCK), (FID_BLOCK)                    ; 1335
                             04  A0  26  A6 3C 0021C    MOVZWL  38(NAME_BLOCK), 4(FID_BLOCK)                    ; 1336
                             08  A0  28  A6 3C 00221    MOVZWL  40(NAME_BLOCK), 8(FID_BLOCK)                    ; 1337
                                 50      58 D0 00226 24$:  MOVL    STATUS, R0                                    ; 1355
                                         04 00229 25$:     RET                                                 ; 1357

; Routine Size:  554 bytes,    Routine Base:  _FDL$CODE + 0000

;  639          1358  1
```

```
;  641     1359  1 %SBTTL  'FDL$PARSE'
;  642     1360  1 GLOBAL ROUTINE  FDL$PARSE =
;  643     1361  1 !++
;  644     1362  1 !
;  645     1363  1 ! Functional Description:
;  646     1364  1 !
;  647     1365  1 !       This routine parses an FDL spec (either file or string) and stuffs
;  648     1366  1 !       the RMS control blocks with the information.
;  649     1367  1 !
;  650     1368  1 ! Calling Sequence:
;  651     1369  1 !
;  652     1370  1 !       fdl$parse(      fdl_spec,
;  653     1371  1 !                       fdl_fab_pointer,
;  654     1372  1 !                       fdl_rab_pointer
;  655     1373  1 !                       [,flags]
;  656     1374  1 !                       [,dflt-fdl-spc]
;  657     1375  1 !                       [,stmnt-num] )
;  658     1376  1 !
;  659     1377  1 ! Input Parameters:
;  660     1378  1 !
;  661     1379  1 !       fdl_spec        - descriptor of the input fdl file name string (required)
;  662     1380  1 !
;  663     1381  1 !       flags           - address of flags longword (optional)
;  664     1382  1 !               FDL$V_SIGNAL            signal errors instead of returning
;  665     1383  1 !               FDL$V_FDL_STRING        input fdl-spec is a char string
;  666     1384  1 !               FDL$V_DEFAULT_STRING    input dflt-fdl-spc is a char string
;  667     1385  1 !               FDL$V_$CALLBACK         used by EDF (FDL$PARSE only)
;  668     1386  1 !
;  669     1387  1 !       dflt-fdl-spc    - descriptor of the default FDL SPEC (optional)
;  670     1388  1 !
;  671     1389  1 ! Implicit Inputs:
;  672     1390  1 !       none
;  673     1391  1 !
;  674     1392  1 ! Output Parameters:
;  675     1393  1 !
;  676     1394  1 !       fab_pointer     - address of a longword to receive the address of the
;  677     1395  1 !                         filled in fab (required)
;  678     1396  1 !
;  679     1397  1 !       rab_pointer     - address of a longword to receive the address of the
;  680     1398  1 !                         filled in rab (required)
;  681     1399  1 !
;  682     1400  1 !       stmnt-num       - address of longword to recieve statement
;  683     1401  1 !                         number (optional)
;  684     1402  1 !
;  685     1403  1 ! Implicit Outputs:
;  686     1404  1 !       none
;  687     1405  1 !
;  688     1406  1 ! Routine Value:
;  689     1407  1 !       none
;  690     1408  1 !
;  691     1409  1 ! Routines Called:
;  692     1410  1 !
;  693     1411  1 !       fdl$$get_vm
;  694     1412  1 !       lib$get_vm
;  695     1413  1 !       lib$free_vm
;  696     1414  1 !
;  697     1415  1 ! Side Effects:
```

```
;  698          1416  1 !       none
;  699          1417  1 !
;  700          1418  1 !--
;  701          1419  1
;  702          1420  2     BEGIN
;  703          1421  2
;  704          1422  2     BUILTIN
;  705          1423  2         ACTUALCOUNT,
;  706          1424  2         ACTUALPARAMETER;
;  707          1425  2
;  708          1426  2     LOCAL
;  709          1427  2         PASS            : BYTE,
;  710          1428  2         COUNT           : BYTE,
;  711          1429  2         LENGTH          : WORD,
;  712          1430  2         ADDR,
;  713          1431  2         FLAGS           : REF BLOCK [ ,BYTE ],
;  714          1432  2         FDL_FAB_PTR,
;  715          1433  2         BYTES           : LONG,
;  716          1434  2         FAB_POINTER,
;  717          1435  2         RAB_POINTER,
;  718          1436  2         STATUS,
;  719          1437  2         USERBUF,
;  720          1438  2         TMP_SPEC        : REF DESC_BLK,
;  721          1439  2         DFLT_SPEC       : REF DESC_BLK,
;  722          1440  2         FDL_SPEC        : REF DESC_BLK;
;  723          1441  2
;  724          1442  2     ! Set up handler to control errors
;  725          1443  2     !
;  726          1444  2     LIB$ESTABLISH ( HANDLER );
;  727          1445  2
;  728          1446  2     ! Clear the valid stmnt flag as a fix...  should always be cleared.
;  729          1447  2     !
;  730          1448  2     FDL$AB_CTRL [ FDL$V_STVALID ] = _CLEAR;
;  731          1449  2
;  732          1450  2
;  733          1451  2     ! Get the number of arguments and see if it's right
;  734          1452  2     !
;  735          1453  2     COUNT = ACTUALCOUNT ();
;  736          1454  2
;  737          1455  3     IF ( ( .COUNT LSS 3 ) OR ( .COUNT GTR 6 ) )
;  738          1456  2     THEN
;  739          1457  2         RETURN FDL$_ILL_ARG;
;  740          1458  2
;  741          1459  2     ! Get the address of the of where the caller wants his FAB and RAB
;  742          1460  2     ! and get the FDL SPEC
;  743          1461  2     !
;  744          1462  2     FDL_SPEC = ACTUALPARAMETER(1);
;  745          1463  2     FAB_POINTER = ACTUALPARAMETER(2);
;  746          1464  2     RAB_POINTER = ACTUALPARAMETER(3);
;  747          1465  2
;  748          1466  2     ! Check 'em out
;  749          1467  2     !
;  750          1468  3     IF (
;  751          1469  4     ( .FAB_POINTER EQLU 0 ) OR ( .RAB_POINTER EQLU 0 ) OR ( .FDL_SPEC EQLU 0 )
;  752          1470  2     ) THEN
;  753          1471  2         RETURN FDL$_ILL_ARG;
;  754          1472  2
```

```
;  755          1473  2      ! Assume we're not doing a default parse.
;  756          1474  2      !
;  757          1475  2      PASS = 0;
;  758          1476  2      TMP_SPEC = .FDL_SPEC;
;  759          1477  2      FDL$AB_CTRL [ FDL$V_DFLT_PRES ] = _CLEAR;
;  760          1478  2      FDL$AB_CTRL [ FDL$V_REPARSE ] = _CLEAR;
;  761          1479  2
;  762          1480  3      IF ( .COUNT GTR 4 )
;  763          1481  2      THEN
;  764          1482  3          BEGIN
;  765          1483  3
;  766          1484  3          DFLT_SPEC = ACTUALPARAMETER (5);
;  767          1485  3
;  768          1486  3          IF .DFLT_SPEC NEQU 0
;  769          1487  3          THEN
;  770          1488  4              BEGIN
;  771          1489  4
;  772          1490  4              FDL$AB_CTRL [ FDL$V_DFLT_PRES ] = _SET;
;  773          1491  4              TMP_SPEC = .DFLT_SPEC;
;  774          1492  4
;  775          1493  3              END;
;  776          1494  3
;  777          1495  2          END;
;  778          1496  2
;  779          1497  2      ! Get the stmnt-num argument (if it hasn't been got already)
;  780          1498  2      !
;  781          1499  2      IF NOT .FDL$AB_CTRL [ FDL$V_STVALID ]
;  782          1500  2      THEN
;  783          1501  3          BEGIN
;  784          1502  3
;  785          1503  3          IF .COUNT GTR 5
;  786          1504  3          THEN
;  787          1505  3              FDL$GL_STNUMPTR = ACTUALPARAMETER (6)
;  788          1506  3          ELSE
;  789          1507  3              FDL$GL_STNUMPTR = 0;
;  790          1508  3
;  791          1509  3          IF .FDL$GL_STNUMPTR NEQU 0
;  792          1510  3          THEN
;  793          1511  3              FDL$AB_CTRL [ FDL$V_STVALID ] = _SET
;  794          1512  3          ELSE
;  795          1513  3              FDL$AB_CTRL [ FDL$V_STVALID ] = _CLEAR;
;  796          1514  3
;  797          1515  2          END;
;  798          1516  2
;  799          1517  2      ! **** THIS IS THE MAJOR LOOP FOR DOING THE DEFAULT PARSE, ****
;  800          1518  2      !                **** AND THEN THE MAIN PARSE. ****
;  801          1519  2      !
;  802          1520  2      DO
;  803          1521  3          BEGIN
;  804          1522  3
;  805          1523  3          ! Keep track of how many times we've passed this way.
;  806          1524  3          !
;  807          1525  3          PASS = .PASS + 1;
;  808          1526  3
;  809          1527  4          IF ( .PASS GTR 1 )
;  810          1528  3          THEN
;  811          1529  3              FDL$AB_CTRL [ FDL$V_REPARSE ] = _SET;
```

```
;  812     1530  3
;  813     1531  3          ! Allow for wierd strings (like VARYING or byte-arrays)
;  814     1532  3          !
;  815     1533  3          RET_ON_ERROR( LIB$ANALYZE_SDESC( .TMP_SPEC,LENGTH,ADDR ) );
;  816     1534  3
;  817     1535  3          ! Setup for reparse (if needed)
;  818     1536  3          !
;  819     1537  3          TMP_SPEC = .FDL_SPEC;
;  820     1538  3
;  821     1539  4          IF ( .LENGTH EQLU 0 ) OR ( .ADDR EQLU 0 )
;  822     1540  3          THEN
;  823     1541  3              RETURN FDL$_ILL_ARG;
;  824     1542  3
;  825     1543  3          FDL$AB_CTRL [ FDL$V_DCL ] = _CLEAR;
;  826     1544  3          FDL$AB_CTRL [ FDL$V_PCALL ] = _CLEAR;
;  827     1545  3          FDL$AB_CTRL [ FDL$V_STRING_SPEC ] = _CLEAR;
;  828     1546  3
;  829     1547  3          ! Get the flags
;  830     1548  3          !
;  831     1549  4          IF ( .COUNT GTR 3 )
;  832     1550  3          THEN
;  833     1551  4              BEGIN
;  834     1552  4
;  835     1553  4              FLAGS = ACTUALPARAMETER (4);
;  836     1554  4
;  837     1555  4              IF .FLAGS NEQU 0
;  838     1556  4              THEN
;  839     1557  5                  BEGIN
;  840     1558  5
;  841     1559  5                  FDL$AB_CTRL [ FDL$V_DCL ] = .FLAGS [ FDL$V_SIGNAL ];
;  842     1560  5                  FDL$AB_CTRL [ FDL$V_PCALL ] = .FLAGS [ FDL$V_$CALLBACK ];
;  843     1561  5
;  844     1562  5                  ! Guard against garbage flags
;  845     1563  5                  !
;  846     1564  6                  IF (
;  847     1565  7                  (.FDL$AB_CTRL [ FDL$V_PCALL ])
;  848     1566  6                  AND
;  849     1567  7                  (.FDL$GL_PCALL EQLU 0)
;  850     1568  5                  ) THEN
;  851     1569  5                      RETURN FDL$_ILL_ARG;
;  852     1570  5
;  853     1571  6                  IF (
;  854     1572  7                  (.FDL$AB_CTRL [ FDL$V_DFLT_PRES ])
;  855     1573  6                  AND
;  856     1574  7                  ( NOT .FDL$AB_CTRL [ FDL$V_REPARSE ])
;  857     1575  5                  ) THEN
;  858     1576  5                      FDL$AB_CTRL [ FDL$V_STRING_SPEC ] =
;  859     1577  5                                              .FLAGS [ FDL$V_DEFAULT_STRING ]
;  860     1578  5                  ELSE
;  861     1579  5                      FDL$AB_CTRL [ FDL$V_STRING_SPEC ] =
;  862     1580  5                                              .FLAGS [ FDL$V_FDL_STRING ];
;  863     1581  5
;  864     1582  4                  END;
;  865     1583  4
;  866     1584  3              END;
;  867     1585  3
;  868     1586  3          ! Setup maximum line size
```

```
;  869          1587  3          !
;  870          1588  3          IF .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
;  871          1589  3          THEN
;  872          1590  3              FDL$GL_MAXLINE = .LENGTH
;  873          1591  3          ELSE
;  874          1592  3              FDL$GL_MAXLINE = FDL$K_MAXLINE;
;  875          1593  3
;  876          1594  3          ! Set up the fab and name block
;  877          1595  3          !
;  878          1596  3          ! Find the number of bytes needed for all of this which is:
;  879          1597  3          ! Space for the fab, name block, expanded and resultant string buffers
;  880          1598  3          ! the user buffer in the rab and the upcase buffer. NOTE: The upcase buffer
;  881          1599  3          ! MUST be allocated directly after the user buffer.
;  882          1600  3          !
;  883          1601  3          ! The space is allocated:
;  884          1602  3          !
;  885          1603  3          !                    ------------------------
;  886          1604  3          !                   |        fab             |  fab$k_bln
;  887          1605  3          !                   |------------------------|
;  888          1606  3          !                   |        nam             |  nam$k_bln
;  889          1607  3          !                   |------------------------|
;  890          1608  3          !                   |  expanded str buffer   |  esa_buf_siz
;  891          1609  3          !                   |------------------------|
;  892          1610  3          !                   | resultant str buffer   |  rsa_buf_siz
;  893          1611  3          !                   |------------------------|
;  894          1612  3          !                   |     user buffer        |  .fdl$gl_maxline
;  895          1613  3          !                   |------------------------|
;  896          1614  3          !                   |    upcase buffer       |  .fdl$gl_maxline
;  897          1615  3          !                    ------------------------
;  898          1616  3          !
;  899          1617  3          !
;  900          1618  3          ! Get the address space. NOTE: Since the user never sees this buffer we
;  901          1619  3          ! can deallocate it when we exit so use lib$get_vm
;  902          1620  3          !
;  903          1621  3          BYTES = FAB$K_BLN + NAM$K_BLN + ESA_BUF_SIZ + RSA_BUF_SIZ +
;  904          1622  3                                                ( .FDL$GL_MAXLINE * 2);
;  905          1623  3
;  906          1624  3          IF NOT LIB$GET_VM( BYTES,FDL_FAB_PTR )
;  907          1625  3          THEN
;  908          1626  3              RETURN FDL$_INSVIRMEM;
;  909          1627  3
;  910          1628  3          ! Zero the space
;  911          1629  3          !
;  912          1630  3          CH$FILL ( 0,.BYTES,.FDL_FAB_PTR );
;  913          1631  3
;  914          1632  3          ! Set up the location of the user buffer
;  915          1633  3          !
;  916          1634  3          USERBUF = .FDL_FAB_PTR + FAB$K_BLN + NAM$K_BLN + ESA_BUF_SIZ +
;  917          1635  3                                                           RSA_BUF_SIZ;
;  918          1636  3
;  919          1637  3          ! Prepare for an fdl-spec whether from a file or a string
;  920          1638  3          !
;  921          1639  3          IF .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
;  922          1640  3          THEN
;  923          1641  4              BEGIN
;  924          1642  4
;  925          1643  4              FDL$AB_FDL_STRING [ DSC$W_LENGTH ] = .LENGTH;
```

```
;  926          1644  4              FDL$AB_FDL_STRING [ DSC$A_POINTER ] = .ADDR
;  927          1645  4
;  928          1646  4              END
;  929          1647  3          ELSE
;  930          1648  4              BEGIN
;  931          1649  4
;  932          1650  4              ! Initialize the FDL FILE fabs and rabs
;  933          1651  4              !
;  934        P 1652  4              $FAB_INIT ( FAB = .FDL_FAB_PTR,
;  935        P 1653  4                          CTX = FDL$_OPENFDL,
;  936        P 1654  4                          DNM = '.FDL',
;  937        P 1655  4                          FAC = GET,
;  938        P 1656  4                          FNS = .LENGTH,
;  939        P 1657  4                          FNA = .ADDR,
;  940        P 1658  4                          FOP = <SQO,NAM>,
;  941          1659  4                          NAM = .FDL_FAB_PTR + FAB$K_BLN );
;  942          1660  4
;  943        P 1661  4              $NAM_INIT ( NAM = .FDL_FAB_PTR + FAB$K_BLN,
;  944        P 1662  4                          ESA = .FDL_FAB_PTR + FAB$K_BLN + NAM$K_BLN,
;  945        P 1663  4                          ESS = ESA_BUF_SIZ,
;  946        P 1664  4                          RSA = .FDL_FAB_PTR + FAB$K_BLN + NAM$K_BLN + ESA_BUF_SIZ,
;  947          1665  4                          RSS = RSA_BUF_SIZ );
;  948          1666  4
;  949        P 1667  4              $RAB_INIT ( RAB = FDL$AB_FDL_RAB,
;  950        P 1668  4                          CTX = FDL$_OPENFDL,
;  951        P 1669  4                          FAB = .FDL_FAB_PTR,
;  952        P 1670  4                          RAC = SEQ,
;  953        P 1671  4                          ROP = RAH,
;  954        P 1672  4                          UBF = .USERBUF,
;  955          1673  4                          USZ = .FDL$GL_MAXLINE );
;  956          1674  4
;  957          1675  4              ! Open the file
;  958          1676  4              !
;  959          1677  4              $OPEN( FAB=.FDL_FAB_PTR,ERR=FDL$$RMS_OPEN_ERROR );
;  960          1678  4
;  961          1679  4              ! Connect a stream
;  962          1680  4              !
;  963          1681  4              $CONNECT( RAB=FDL$AB_FDL_RAB,ERR=FDL$$RMS_OPEN_ERROR );
;  964          1682  4
;  965          1683  4              ! Errors from now on are read errors
;  966          1684  4              !
;  967          1685  4              FDL$AB_FDL_RAB [ RAB$L_CTX ] = FDL$_READERR
;  968          1686  4
;  969          1687  3              END;
;  970          1688  3
;  971          1689  4          IF ( .PASS EQLU 1 )
;  972          1690  3          THEN
;  973          1691  4              BEGIN
;  974          1692  4
;  975          1693  4              ! Get the address space for the Parsed-fab and the Parsed-rab
;  976          1694  4              !
;  977          1695  4              FDL$AB_PARSED_FAB = FDL$$GET_VM( FAB$K_BLN );
;  978          1696  4              FDL$AB_PARSED_RAB = FDL$$GET_VM( RAB$K_BLN );
;  979          1697  4
;  980          1698  4              ! Init the Parsed-fab and Parsed-rab (Set some defaults as well)
;  981          1699  4              !
;  982        P 1700  4              $FAB_INIT ( FAB = .FDL$AB_PARSED_FAB,
```

```
;  983        P 1701  4                        FOP = <MXV,NAM>,
;  984        P 1702  4                        ORG = SEQ,
;  985        P 1703  4                        RAT = CR,
;  986          1704  4                        RFM = VAR );
;  987          1705  4
;  988        P 1706  4              $RAB_INIT ( RAB = .FDL$AB_PARSED_RAB,
;  989          1707  4                          FAB = .FDL$AB_PARSED_FAB );
;  990          1708  4
;  991          1709  4              ! Give the fab and rab to the user
;  992          1710  4              !
;  993          1711  4              .FAB_POINTER = .FDL$AB_PARSED_FAB;
;  994          1712  4              .RAB_POINTER = .FDL$AB_PARSED_RAB;
;  995          1713  4
;  996          1714  3              END;
;  997          1715  3
;  998          1716  3          ! Initialize the line buffer and the upcase buffer
;  999          1717  3          !
; 1000          1718  3          FDL$AB_LINE [ DSC$A_POINTER ] = .USERBUF;
; 1001          1719  3
; 1002          1720  3          FDL$AB_UPCASED [ DSC$A_POINTER ] = .FDL$AB_LINE [ DSC$A_POINTER ] +
; 1003          1721  3                                                             .FDL$GL_MAXLINE;
; 1004          1722  3
; 1005          1723  3          ! Initialize the parser
; 1006          1724  3          !
; 1007          1725  3          FDL$$INIT_PARSE();
; 1008          1726  3
; 1009          1727  3          ! Parse the file
; 1010          1728  3          !
; 1011        P 1729  3          RET_ON_ERROR (
; 1012          1730  3          LIB$TPARSE( FDL$AB_TPARSE_BLOCK,FDL$AB_STATE_TABLE,FDL$AB_KEY_TABLE ));
; 1013          1731  3
; 1014          1732  3          STATUS = FDL$$FINISH_PARSE();
; 1015          1733  3
; 1016          1734  3          IF NOT .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
; 1017          1735  3          THEN
; 1018          1736  4              BEGIN
; 1019          1737  4
; 1020          1738  4              ! Close down the FDL file and exit
; 1021          1739  4              !
; 1022          1740  4              $DISCONNECT( RAB=FDL$AB_FDL_RAB );
; 1023          1741  4
; 1024          1742  4              $CLOSE( FAB=.FDL_FAB_PTR );
; 1025          1743  4
; 1026          1744  4                  ! Deallocate the memory for the fab etc
; 1027          1745  4                  !
; 1028          1746  5                  BEGIN
; 1029          1747  5                  LOCAL STATUS;
; 1030          1748  5
; 1031          1749  6                  IF NOT ( STATUS = LIB$FREE_VM( BYTES,FDL_FAB_PTR ))
; 1032          1750  5                  THEN
; 1033          1751  5                      SIGNAL_STOP ( .STATUS );
; 1034          1752  4                  END;
; 1035          1753  4
; 1036          1754  3              END;
; 1037          1755  3
; 1038          1756  3          END      ! OF MAJOR PARSE/REPARSE LOOP
; 1039          1757  3
```

```
;  1040          1758  3     UNTIL (
;  1041          1759  4     ( NOT .FDL$AB_CTRL [ FDL$V_DFLT_PRES ] )
;  1042          1760  3     OR
;  1043          1761  4     ( .FDL$AB_CTRL [ FDL$V_DFLT_PRES ] AND .FDL$AB_CTRL [ FDL$V_REPARSE ] )
;  1044          1762  2     );
;  1045          1763  2
;  1046          1764  2     RETURN .STATUS
;  1047          1765  2
;  1048          1766  1     END;

                                                        .PSECT  _FDL$PLIT,NOWRT,NOEXE,  SHR,  PIC,2

                                              4C  44  46  2E  00000 P.AAA:  .ASCII  \.FDL\                                  ;

                                                        $RMS_PTR=           FDL$AB_FDL_RAB
                                                                .EXTRN  SYS$OPEN, SYS$CONNECT
                                                                .EXTRN  SYS$DISCONNECT

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

                                                  0FFC 00000            .ENTRY  FDL$PARSE, Save R2,R3,R4,R5,R6,R7,R8,R9,-     ; 1360
                                                                                R10,R11
                                       5E                28  C2 00002            SUBL2   #40, SP                               ;
                                            00000000V    00  9F 00005            PUSHAB  HANDLER                               ; 1444
                                 00000000G  00           01  FB 0000B            CALLS   #1, LIB$ESTABLISH                     ;
                                 00000000G  00           04  8A 00012            BICB2   #4, FDL$AB_CTRL+2                     ; 1448
                                       5B                6C  90 00019            MOVB    (AP), COUNT                           ; 1453
                                       03                5B  91 0001C            CMPB    COUNT, #3                             ; 1455
                                                         03  1E 0001F            BGEQU   2$                                    ;
                                                     00E0  31 00021 1$:          BRW     10$                                   ;
                                       06                5B  91 00024 2$:        CMPB    COUNT, #6                             ;
                                                         F8  1A 00027            BGTRU   1$                                    ;
                                       6E          04    AC  D0 00029            MOVL    4(AP), FDL_SPEC                       ; 1462
                                    10 AE          08    AC  D0 0002D            MOVL    8(AP), FAB_POINTER                    ; 1463
                                    0C AE          0C    AC  D0 00032            MOVL    12(AP), RAB_POINTER                   ; 1464
                                                   10    AE  D5 00037            TSTL    FAB_POINTER                           ; 1469
                                                         F5  13 0003A            BEQL    1$                                    ;
                                                   0C    AE  D5 0003C            TSTL    RAB_POINTER                           ;
                                                         E0  13 0003F            BEQL    1$                                    ;
                                                         6E  D5 00041            TSTL    FDL_SPEC                              ;
                                                         DC  13 00043            BEQL    1$                                    ;
                                                   08    AE  94 00045            CLRB    PASS                                  ; 1475
                                          04 AE          6E  D0 00048            MOVL    FDL_SPEC, TMP_SPEC                    ; 1476
                                 00000000G  00           03  8A 0004C            BICB2   #3, FDL$AB_CTRL+2                     ; 1478
                                       04                5B  91 00053            CMPB    COUNT, #4                             ; 1480
                                                         11  1B 00056            BLEQU   3$                                    ;
                                       50          14    AC  D0 00058            MOVL    20(AP), DFLT_SPEC                     ; 1484
                                                         0B  13 0005C            BEQL    3$                                    ; 1486
                                 00000000G  00           02  88 0005E            BISB2   #2, FDL$AB_CTRL+2                     ; 1490
                                          04 AE          50  D0 00065            MOVL    DFLT_SPEC, TMP_SPEC                   ; 1491
                              27 00000000G  00           02  E0 00069 3$:        BBS     #2, FDL$AB_CTRL+2, 7$                 ; 1499
                                       05                5B  91 00071            CMPB    COUNT, #5                             ; 1503
                                                         0A  1B 00074            BLEQU   4$                                    ;
                                 00000000G  00     18    AC  D0 00076            MOVL    24(AP), FDL$GL_STNUMPTR               ; 1505
                                                         06  11 0007E            BRB     5$                                    ;
```

```
                                                00000000G  00  D4 00080 4$:     CLRL    FDL$GL_STNUMPTR                                 ; 1507
                                                           09  13 00086 5$:     BEQL    6$                                              ; 1509
                                  00000000G  00            04  88 00088       BISB2   #4, FDL$AB_CTRL+2                               ; 1511
                                                           07  11 0008F       BRB     7$                                              ;
                                  00000000G  00            04  8A 00091 6$:     BICB2   #4, FDL$AB_CTRL+2                           ; 1513
                                                 08        AE  96 00098 7$:     INCB    PASS                                            ; 1525
                                             01  08        AE  91 0009B       CMPB    PASS, #1                                        ; 1527
                                                           07  1B 0009F       BLEQU   8$                                              ;
                                  00000000G  00            01  88 000A1       BISB2   #1, FDL$AB_CTRL+2                           ; 1529
                                                 18        AE  9F 000A8 8$:     PUSHAB  ADDR                                            ; 1533
                                                 20        AE  9F 000AB       PUSHAB  LENGTH                                          ;
                                                 0C        AE  DD 000AE       PUSHL   TMP_SPEC                                        ;
                                  00000000G  00            03  FB 000B1       CALLS   #3, LIB$ANALYZE_SDESC                           ;
                                             01            50  E8 000B8       BLBS    STATUS, 9$                                      ;
|                                                          04  000BB       RET                                                     ;
|                                       04   AE            6E  D0 000BC 9$:     MOVL    FDL_SPEC, TMP_SPEC                          ; 1537
                                             59  1C        AE  3C 000C0       MOVZWL  LENGTH, R9                                      ; 1539
|                                                          3E  13 000C4       BEQL    10$                                             ;
|                                                 18        AE  D5 000C6       TSTL    ADDR                                            ;
|                                                          39  13 000C9       BEQL    10$                                             ;
|                                  00000000G  00            1C  8A 000CB       BICB2   #28, FDL$AB_CTRL+1                          ; 1545
|                                             03            5B  91 000D2       CMPB    COUNT, #3                                       ; 1549
|                                                          59  1B 000D5       BLEQU   14$                                             ;
|                                             5A  10        AC  D0 000D7       MOVL    16(AP), FLAGS                               ; 1553
|                                                          53  13 000DB       BEQL    14$                                             ; 1555
| 00000000G  00            01                03            6A  F0 000DD       INSV    (FLAGS), #3, #1, FDL$AB_CTRL+1              ; 1559
|            50            6A                01            04  EF 000E6       EXTZV   #4, #1, (FLAGS), R0                         ; 1560
| 00000000G  00            01                02            50  F0 000EB       INSV    R0, #2, #1, FDL$AB_CTRL+1                   ;
|                       10 00000000G  00                   02  E1 000F4       BBC     #2, FDL$AB_CTRL+1, 11$                      ; 1565
|                                     00000000G  00  D5 000FC       TSTL    FDL$GL_PCACL                                    ; 1567
|                                                08  12 00102       BNEQ    11$                                             ;
|                                             50 00000000G  8F  D0 00104 10$:     MOVL    #FDL$_ILL_ARG, R0                           ; 1569
|                                                          04  0010B       RET                                                     ;
|                       0E 00000000G  00                   01  E1 0010C 11$:     BBC     #1, FDL$AB_CTRL+2, 12$                      ; 1572
|                                     07 00000000G  00  E8 00114       BLBS    FDL$AB_CTRL+2, 12$                              ; 1574
|            50            6A                01            02  EF 0011B       EXTZV   #2, #1, (FLAGS), R0                         ; 1577
|                                                          05  11 00120       BRB     13$                                             ;
|            50            6A                01            01  EF 00122 12$:     EXTZV   #1, #1, (FLAGS), R0                         ; 1580
| 00000000G  00            01                04            50  F0 00127 13$:     INSV    R0, #4, #1, FDL$AB_CTRL+1                   ;
|                       09 00000000G  00                   04  E1 00130 14$:     BBC     #4, FDL$AB_CTRL+1, 15$                      ; 1588
|                          00000000G  00                   59  D0 00138       MOVL    R9, FDL$GL_MAXLINE                          ; 1590
|                                                          09  11 0013F       BRB     16$                                             ;
|                          00000000G  00        0400       8F  3C 00141 15$:     MOVZWL  #1024, FDL$GL_MAXLINE                       ; 1592
|                                             50 00000000G  00  D0 0014A 16$:     MOVL    FDL$GL_MAXLINE, R0                          ; 1622
|                  24   AE                    50            01  78 00151       ASHL    #1, R0, BYTES                               ; 1621
|                                       24   AE 000002AE    8F  C0 00156       ADDL2   #686, BYTES                                 ;
|                                                 20        AE  9F 0015E       PUSHAB  FDL_FAB_PTR                                 ; 1624
|                                                 28        AE  9F 00161       PUSHAB  BYTES                                           ;
|                          00000000G  00                   02  FB 00164       CALLS   #2, LIB$GET_VM                                  ;
|                                                          08  50  E8 0016B       BLBS    R0, 17$                                         ;
|                                             50 00000000G  8F  D0 0016E       MOVL    #FDL$_INSVIRMEM, R0                         ; 1626
|                                                          04  00175       RET                                                     ;
|                                             58        20  AE  D0 00176 17$:     MOVL    FDL_FAB_PTR, R8                             ; 1630
|       24   AE                    00            6E            00  2C 0017A       MOVC5   #0, (SP), #0, BYTES, (R8)                   ;
|                                                          68      00180                                                           ;
|                                             56      02AE  C8  9E 00181       MOVAB   686(R8), USERBUF                            ; 1634
|                       12 00000000G  00                   04  E1 00186       BBC     #4, FDL$AB_CTRL+1, 18$                      ; 1639
```

```
                    00000000G  00            59  B0 0018E         MOVW    R9, FDL$AB_FDL_STRING                     ; 1643
                    00000000G  00       18   AE  D0 00195         MOVL    ADDR, FDL$AB_FDL_STRING+4                 ; 1644
                                       00D8  31 0019D         BRW     19$
0050   8F        00            6E            00  2C 001A0 18$:    MOVC5   #0, (SP), #0, #80, (R8)                   ; 1659
                                             68     001A7                                                       ;
                               68     5003   8F  B0 001A8         MOVW    #20483, (R8)                              ;
                          04   A8 01000040   8F  D0 001AD         MOVL    #16777280, 4(R8)                          ;
                          16   A8            02  90 001B5         MOVB    #2, 22(R8)                                ;
                          18   A8 00000000G  8F  D0 001B9         MOVL    #FDL$_OPENFDL, 24(R8)                     ;
                          1F   A8            02  90 001C1         MOVB    #2, 31(R8)                                ;
                               57       50   A8  9E 001C5         MOVAB   80(R8), R7                                ;
                          28   A8            57  D0 001C9         MOVL    R7, 40(R8)                                ;
                          2C   A8       18   AE  D0 001CD         MOVL    ADDR, 44(R8)                              ;
                          30   A8 00000000'  00  9E 001D2         MOVAB   P.AAA, 48(R8)                             ;
                          34   A8            59  90 001DA         MOVB    R9, 52(R8)                                ;
                          35   A8            04  90 001DE         MOVB    #4, 53(R8)                                ;
0060   8F        00            6E            00  2C 001E2         MOVC5   #0, (SP), #0, #96, (R7)                   ; 1665
                                             67     001E9                                                       ;
                               67     6002   8F  B0 001EA         MOVW    #24578, (R7)                              ;
                          02   A7            01  8E 001EF         MNEGB   #1, 2(R7)                                 ;
                          04   A7     01AF   C8  9E 001F3         MOVAB   431(R8), 4(R7)                            ;
                          0A   A7            01  8E 001F9         MNEGB   #1, 10(R7)                                ;
                          0C   A7     00B0   C8  9E 001FD         MOVAB   176(R8), 12(R7)                           ;
0044   8F        00            6E            00  2C 00203         MOVC5   #0, (SP), #0, #68, $RMS_PTR               ; 1673
                                  00000000'  00     0020A                                                       ;
                    00000000'  00     4401   8F  B0 0020F         MOVW    #17409, $RMS_PTR                          ;
                    00000000'  00     0200   8F  3C 00218         MOVZWL  #512, $RMS_PTR+4                          ;
                    00000000'  00 00000000G  8F  D0 00221         MOVL    #FDL$_OPENFDL, $RMS_PTR+24                ;
                                  00000000'  00  94 0022C         CLRB    $RMS_PTR+30                               ;
                    00000000'  00 00000000G  00  B0 00232         MOVW    FDL$GL_MAXLINE, $RMS_PTR+32               ;
                    00000000'  00            56  D0 0023D         MOVL    USERBUF, $RMS_PTR+36                      ;
                    00000000'  00            58  D0 00244         MOVL    R8, $RMS_PTR+60                           ;
                                  00000000G  00  9F 0024B         PUSHAB  FDL$$RMS_OPEN_ERROR                       ; 1677
                                             58  DD 00251         PUSHL   R8                                        ;
                    00000000G  00            02  FB 00253         CALLS   #2, SYS$OPEN                              ;
                                  00000000G  00  9F 0025A         PUSHAB  FDL$$RMS_OPEN_ERROR                       ; 1681
                                  00000000'  00  9F 00260         PUSHAB  FDL$AB_FDL_RAB                            ;
                    00000000G  00            02  FB 00266         CALLS   #2, SYS$CONNECT                           ;
                    00000000'  00 00000000G  8F  D0 0026D         MOVL    #FDL$_READERR, FDL$AB_FDL_RAB+24          ; 1685
                               01       08   AE  91 00278 19$:    CMPB    PASS, #1                                  ; 1689
                                             6E  12 0027C         BNEQ    20$                                       ;
                               7E       50   8F  9A 0027E         MOVZBL  #80, -(SP)                                ; 1695
                    00000000G  00            01  FB 00282         CALLS   #1, FDL$$GET_VM                           ;
                    00000000'  00            50  D0 00289         MOVL    R0, FDL$AB_PARSED_FAB                     ;
                               7E       44   8F  9A 00290         MOVZBL  #68, -(SP)                                ; 1696
                    00000000G  00            01  FB 00294         CALLS   #1, FDL$$GET_VM                           ;
                    00000000'  00            50  D0 0029B         MOVL    R0, FDL$AB_PARSED_RAB                     ;
                               57 00000000'  00  D0 002A2         MOVL    FDL$AB_PARSED_FAB, R7                     ; 1704
0050   8F        00            6E            00  2C 002A9         MOVC5   #0, (SP), #0, #80, (R7)                   ;
                                             67     002B0                                                       ;
                               67     5003   8F  B0 002B1         MOVW    #20483, (R7)                              ;
                          04   A7 01000002   8F  D0 002B6         MOVL    #16777218, 4(R7)                          ;
                          16   A7            02  90 002BE         MOVB    #2, 22(R7)                                ;
                          1D   A7     0200   8F  B0 002C2         MOVW    #512, 29(R7)                              ;
                          1F   A7            02  90 002C8         MOVB    #2, 31(R7)                                ;
                               59 00000000'  00  D0 002CC         MOVL    FDL$AB_PARSED_RAB, R9                     ; 1707
0044   8F        00            6E            00  2C 002D3         MOVC5   #0, (SP), #0, #68, (R9)                   ;
```

```
                                              69  002DA
                          69      4401       8F B0 002DB            MOVW    #17409, (R9)                               ;
                    3C    A9                 57 D0 002E0            MOVL    R7, 60(R9)                                 ;
                    10    BE                 57 D0 002E4            MOVL    R7, @FAB_POINTER                           ; 1711
                    0C    BE                 59 D0 002E8            MOVL    R9, @RAB_POINTER                           ; 1712
             00000000G    00                 56 D0 002EC  20$:      MOVL    USERBUF, FDL$AB_LINE+4                     ; 1718
 00000000G  00 00000000G  00 00000000G       00 C1 002F3            ADDL3   FDL$GL_MAXLINE, FDL$AB_LINE+4, -           ; 1721
                                                                            FDL$AB_UPCASED+4                           ;
             00000000G    00                 00 FB 00303            CALLS   #0, FDL$$INIT_PARSE                        ; 1725
                             00000000G       00 9F 0030A            PUSHAB  FDL$AB_KEY_TABLE                           ; 1730
                             00000000G       00 9F 00310            PUSHAB  FDL$AB_STATE_TABLE                         ;
                             00000000G       00 9F 00316            PUSHAB  FDL$AB_TPARSE_BLOCK                        ;
             00000000G    00                 03 FB 0031C            CALLS   #3, LIB$TPARSE                             ;
                          5C                 50 E9 00323            BLBC    STATUS, 23$                                ;
             00000000G    00                 00 FB 00326            CALLS   #0, FDL$$FINISH_PARSE                      ; 1732
                    14    AE                 50 D0 0032D            MOVL    R0, STATUS                                 ;
          2F 00000000G    00                 04 E0 00331            BBS     #4, FDL$AB_CTRL+1, 21$                     ; 1734
                             00000000'       00 9F 00339            PUSHAB  FDL$AB_FDL_RAB                             ; 1740
             00000000G    00                 01 FB 0033F            CALLS   #1, SYS$DISCONNECT                         ;
                                             58 DD 00346            PUSHL   R8                                         ; 1742
             00000000G    00                 01 FB 00348            CALLS   #1, SYS$CLOSE                              ;
                                      20     AE 9F 0034F            PUSHAB  FDL_FAB_PTR                                ; 1749
                                      28     AE 9F 00352            PUSHAB  BYTES                                      ;
             00000000G    00                 02 FB 00355            CALLS   #2, LIB$FREE_VM                            ;
                          09                 50 E8 0035C            BLBS    STATUS, 21$                                ;
                                             50 DD 0035F            PUSHL   STATUS                                     ; 1751
             00000000G    00                 01 FB 00361            CALLS   #1, LIB$STOP                               ;
 50 00000000G  00         01                 01 EF 00368  21$:      EXTZV   #1, #1, FDL$AB_CTRL+2, R0                  ; 1759
                          0A                 50 E9 00371            BLBC    R0, 22$                                    ;
                          03 00000000G       00 E8 00374            BLBS    FDL$AB_CTRL+2, 22$                         ; 1761
                                        FD1A 31 0037B            BRW     7$                                         ;
                          50      14         AE D0 0037E  22$:      MOVL    STATUS, R0                                 ; 1764
                                                04 00382  23$:      RET                                                ; 1766

; Routine Size. 899 bytes,    Routine Base:  _FDL$CODE + 022A
```

```
;                       1767  1 %SBTTL  'FDL$RELEASE'
;                       1768  1 GLOBAL ROUTINE  FDL$RELEASE =
;                       1769  1 !++
;                       1770  1 !
;                       1771  1 ! Functional Description:
;                       1772  1 !
;                       1773  1 !       This routine takes a FAB address or a RAB address or both and
;                       1774  1 !       tracks down the RMS control block chains and deallocates them
;                       1775  1 !       for the user.
;                       1776  1 !
;                       1777  1 ! Calling Sequence:
;                       1778  1 !
;                       1779  1 !       fdl$release (   [fab_pointer]
;                       1780  1 !                       [,rab_pointer]
;                       1781  1 !                       [,flags]
;                       1782  1 !                       [,badblk-addr] )
;                       1783  1 !
;                       1784  1 ! Input Parameters:
;                       1785  1 !
;                       1786  1 !       fab_pointer     - address of a longword containing the address
;                       1787  1 !                         of the FAB to deallocate (plus attachments)
;                       1788  1 !
;                       1789  1 !       rab_pointer     - address of a longword containing the address
;                       1790  1 !                         of the RAB to deallocate (plus attachments)
;                       1791  1 !
;                       1792  1 !       flags           - address of flags longword (optional)
;                       1793  1 !               FDL$V_SIGNAL            signal errors instead of returning
;                       1794  1 !
;                       1795  1 ! Implicit Inputs:
;                       1796  1 !       none
;                       1797  1 !
;                       1798  1 ! Output Parameters:
;                       1799  1 !
;                       1800  1 !       badblk-addr     - address of a longword to receive the address
;                       1801  1 !                         of the RMS control block that is invalid (optional)
;                       1802  1 !
;                       1803  1 ! Implicit Outputs:
;                       1804  1 !       none
;                       1805  1 !
;                       1806  1 ! Routine Value:
;                       1807  1 !       none
;                       1808  1 !
;                       1809  1 ! Routines Called:
;                       1810  1 !
;                       1811  1 !       fdl$$check_blocks
;                       1812  1 !
;                       1813  1 ! Side Effects:
;                       1814  1 !       none
;                       1815  1 !
;                       1816  1 !--
;                       1817  1
;                       1818  2     BEGIN
;                       1819  2
;                       1820  2     BUILTIN
;                       1821  2         ACTUALCOUNT,
;                       1822  2         ACTUALPARAMETER;
;                       1823  2
```

```
;  1107     1824  2      LOCAL
;  1108     1825  2          FAB_POINTER             : REF BLOCK [ ,BYTE ],
;  1109     1826  2          RAB_POINTER             : REF BLOCK [ ,BYTE ],
;  1110     1827  2          XAB_POINTER             : REF BLOCK [ ,BYTE ],
;  1111     1828  2          NAM_POINTER             : REF BLOCK [ ,BYTE ],
;  1112     1829  2          SAVE_POINTER            : REF BLOCK [ ,BYTE ],
;  1113     1830  2          FLAGS                   : REF BLOCK [ ,BYTE ],
;  1114     1831  2          STATUS                  : LONG,
;  1115     1832  2          COUNT                   : LONG;
;  1116     1833  2
;  1117     1834  2      ! Setup to handle ACCVIOs and other trash
;  1118     1835  2      !
;  1119     1836  2      LIB$ESTABLISH ( HANDLER );
;  1120     1837  2
;  1121     1838  2      ! Validate the arguments.
;  1122     1839  2      !
;  1123     1840  2      COUNT = ACTUALCOUNT ();
;  1124     1841  2
;  1125     1842  2      ! Clear the valid stmnt flag as a fix...  should always be cleared on entry
;  1126     1843  2      !
;  1127     1844  2      FDL$AB_CTRL [ FDL$V_STVALID ] = _CLEAR;
;  1128     1845  2
;  1129     1846  3      IF ( .COUNT LSS 1 ) OR ( .COUNT GTR 4 )
;  1130     1847  2      THEN
;  1131     1848  2          RETURN FDL$_ILL_ARG;
;  1132     1849  2
;  1133     1850  2      ! 1st arg is address of longword containing address of FAB
;  1134     1851  2      !
;  1135     1852  2      FAB_POINTER = ACTUALPARAMETER (1);
;  1136     1853  2
;  1137     1854  2      ! 2nd arg is address of longword containing address of RAB
;  1138     1855  2      !
;  1139     1856  2      IF .COUNT GTR 1
;  1140     1857  2      THEN
;  1141     1858  2          RAB_POINTER = ACTUALPARAMETER (2)
;  1142     1859  2      ELSE
;  1143     1860  2          RAB_POINTER = 0;
;  1144     1861  2
;  1145     1862  2      ! 3rd arg is address of flags longword
;  1146     1863  2      !
;  1147     1864  2      IF .COUNT GTR 2
;  1148     1865  2      THEN
;  1149     1866  3          BEGIN
;  1150     1867  3
;  1151     1868  3          FLAGS = ACTUALPARAMETER (3);
;  1152     1869  3
;  1153     1870  3          IF .FLAGS NEQU 0
;  1154     1871  3          THEN
;  1155     1872  3              FDL$AB_CTRL [ FDL$V_DCL ] = .FLAGS [ FDL$V_SIGNAL ]
;  1156     1873  3          ELSE
;  1157     1874  3              FDL$AB_CTRL [ FDL$V_DCL ] = _CLEAR;
;  1158     1875  3
;  1159     1876  2          END;
;  1160     1877  2
;  1161     1878  2      ! 4th arg is address of longword to receive address of bad block
;  1162     1879  2      !
;  1163     1880  2      IF .COUNT GTR 3
```

```
;  1164        1881  2      THEN
;  1165        1882  2          FDL$GL_INVBLK_PTR = ACTUALPARAMETER (4)
;  1166        1883  2      ELSE
;  1167        1884  2          FDL$GL_INVBLK_PTR = 0;
;  1168        1885  2
;  1169        1886  2      ! Get actual address of FAB
;  1170        1887  2      !
;  1171        1888  2      IF .FAB_POINTER NEQU 0
;  1172        1889  2      THEN
;  1173        1890  2          FAB_POINTER = ..FAB_POINTER;
;  1174        1891  2
;  1175        1892  2      ! Get actual address of RAB
;  1176        1893  2      !
;  1177        1894  2      IF .RAB_POINTER NEQU 0
;  1178        1895  2      THEN
;  1179        1896  2          RAB_POINTER = ..RAB_POINTER;
;  1180        1897  2
;  1181        1898  2      ! Go check and deallocate the ctrl blocks
;  1182        1899  2      !
;  1183        1900  2      FDL$AB_CTRL [ FDL$V_DEALLOC ] = _SET;
;  1184        1901  2      FDL$$CHECK_BLOCKS ( .FAB_POINTER, .RAB_POINTER );
;  1185        1902  2
;  1186        1903  2      RETURN SS$_NORMAL;
;  1187        1904  2
;  1188        1905  1      END;


                                                003C 00000          .ENTRY  FDL$RELEASE, Save R2,R3,R4,R5                   ; 1768
                                    55 00000000G    00   9E 00002    MOVAB   FDL$GL_INVBLK_PTR, R5                           ;
                                    54 00000000G    00   9E 00009    MOVAB   FDL$AB_CTRL, R4                                 ;
                                       00000000V    00   9F 00010    PUSHAB  HANDLER                                         ; 1836
                          00000000G 00              01   FB 00016    CALLS   #1, LIB$ESTABLISH                               ;
                                    51              6C   9A 0001D    MOVZBL  (AP), COUNT                                     ; 1840
                               02   A4              04   8A 00020    BICB2   #4, FDL$AB_CTRL+2                               ; 1844
                                                   51   D5 00024    TSTL    COUNT                                           ; 1846
                                                   05   15 00026    BLEQ    1$                                              ;
                                    04             51   D1 00028    CMPL    COUNT, #4                                       ;
                                                   08   15 0002B    BLEQ    2$                                              ;
                                    50 00000000G   8F   D0 0002D 1$: MOVL    #FDL$_ILL_ARG, R0                               ; 1848
                                                   04   00034       RET                                                     ;
                                    53        04   AC   D0 00035 2$: MOVL    4(AP), FAB_POINTER                              ; 1852
                                    01             51   D1 00039    CMPL    COUNT, #1                                       ; 1856
                                                   06   15 0003C    BLEQ    3$                                              ;
                                    52        08   AC   D0 0003E    MOVL    8(AP), RAB_POINTER                              ; 1858
                                                   02   11 00042    BRB     4$                                              ;
                                                   52   D4 00044 3$: CLRL    RAB_POINTER                                     ; 1860
                                    02             51   D1 00046 4$: CMPL    COUNT, #2                                       ; 1864
                                                   12   15 00049    BLEQ    6$                                              ;
                                    50        0C   AC   D0 0004B    MOVL    12(AP), FLAGS                                   ; 1868
                                                   08   13 0004F    BEQL    5$                                              ; 1870
01   A4                      01        03          60   F0 00051    INSV    (FLAGS), #3, #1, FDL$AB_CTRL+1                  ; 1872
                                                   04   11 00057    BRB     6$                                              ;
                               01   A4              08   8A 00059 5$: BICB2   #8, FDL$AB_CTRL+1                           ; 1874
                                    03             51   D1 0005D 6$: CMPL    COUNT, #3                                       ; 1880
```

```
                                 06  15 00060          BLEQ    7$                                                      ;
                          65     10  AC  D0 00062          MOVL    16(AP), FDL$GL_INVBLK_PTR                           ; 1882
                                 02  11 00066          BRB     8$                                                      ;
                                 65  D4 00068 7$:      CLRL    FDL$GL_INVBLK_PTR                                   ; 1884
                                 53  D5 0006A 8$:      TSTL    FAB_POINTER                                         ; 1888
                                 03  13 0006C          BEQL    9$                                                      ;
                          53         63  D0 0006E          MOVL    (FAB_POINTER), FAB_POINTER                          ; 1890
                                 52  D5 00071 9$:      TSTL    RAB_POINTER                                         ; 1894
                                 03  13 00073          BEQL    10$                                                     ;
                          52         62  D0 00075          MOVL    (RAB_POINTER), RAB_POINTER                          ; 1896
                       02    A4        20  88 00078 10$:     BISB2   #32, FDL$AB_CTRL+2                          ; 1900
                                 52  DD 0007C          PUSHL   RAB_POINTER                                         ; 1901
                                 53  DD 0007E          PUSHL   FAB_POINTER                                         ;
                 00000000G       00  02  FB 00080          CALLS   #2, FDL$$CHECK_BLOCKS                               ;
                          50         01  D0 00087          MOVL    #1, R0                                              ; 1903
                                     04 0008A          RET                                                         ; 1905

; Routine Size:  139 bytes,    Routine Base:  _FDL$CODE + 05AD
```

```
;1190     1906  1 %SBTTL  'FDL$GENERATE'
;1191     1907  1 GLOBAL ROUTINE  FDL$GENERATE =
;1192     1908  1 !++
;1193     1909  1 !
;1194     1910  1 ! Functional Description:
;1195     1911  1 !
;1196     1912  1 !       This routine produces an FDL SPEC. It is placed either in a char string,
;1197     1913  1 !       or in an FDL file. EVERY RMS FIELD IS INSPECTED AND OUTPUT (if the flag
;1198     1914  1 !       bit FDL$V_FULL_OUTPUT is set).
;1199     1915  1 !       This is essentially a picture of the RMS control blocks.
;1200     1916  1 !
;1201     1917  1 ! Calling Sequence:
;1202     1918  1 !
;1203     1919  1 ! fdl$generate (  flags,
;1204     1920  1 !                  fab_pointer,
;1205     1921  1 !                  rab_pointer
;1206     1922  1 !                [,fdl_file_dst]
;1207     1923  1 !                [,fdl_file_resnam]
;1208     1924  1 !                [,fdl_str_dst]
;1209     1925  1 !                [,bad_blk_addr]
;1210     1926  1 !                [,ret[en]] )
;1211     1927  1 !
;1212     1928  1 ! Input Parameters:
;1213     1929  1 !
;1214     1930  1 !       flags           - address of flags longword (optional)
;1215     1931  1 !               FDL$V_SIGNAL            signal errors instead of returning
;1216     1932  1 !               FDL$V_FDL_STRING        output fdl-spec is a char string
;1217     1933  1 !               FDL$V_FUL[_OUTPUT       all RMS fields are to be output
;1218     1934  1 !               FDL$V_$CAL[BACK         used by EDF
;1219     1935  1 !
;1220     1936  1 !       fab_pointer     - address of a longword which holds the address of
;1221     1937  1 !                         the FAB to be scanned
;1222     1938  1 !
;1223     1939  1 !       rab_pointer     - address of a longword which holds the address of
;1224     1940  1 !                         the RAB to be scanned
;1225     1941  1 !
;1226     1942  1 ! Implicit Inputs:
;1227     1943  1 !       none
;1228     1944  1 !
;1229     1945  1 ! Output Parameters:
;1230     1946  1 !
;1231     1947  1 !       fdl_file_dst    - address of a string descriptor which holds the
;1232     1948  1 !                         filename of the FDL file to be output
;1233     1949  1 !                         (required if FDL$V_FDL_STRING bit is NOT set,
;1234     1950  1 !                          ignored otherwise)
;1235     1951  1 !
;1236     1952  1 !       fdl_file_resnam - address of a dynamic or varying string descriptor
;1237     1953  1 !                         which will hold the result name of the FDL file
;1238     1954  1 !                         that may be created by FDL$GENERATE (optional)
;1239     1955  1 !
;1240     1956  1 !       fdl_str_dst     - address of a dynamic or varying string descriptor
;1241     1957  1 !                         which will hold the fdl spec as a char string
;1242     1958  1 !                         (required if FDL$V_FDL_STRING bit is set,
;1243     1959  1 !                          ignored otherwise)
;1244     1960  1 !
;1245     1961  1 !       bad_blk_addr    - address of a longword to receive the address of
;1246     1962  1 !                         an invalid RMS control block, if detected (optional)
```

```
;1247     1963  1 !
;1248     1964  1 !       retlen          - address of longword to recieve length of the returned
;1249     1965  1 !                         string: either the fdl_file_resnam or the fdl_str_dst
;1250     1966  1 !                         (optional)
;1251     1967  1 !
;1252     1968  1 ! Implicit Outputs:
;1253     1969  1 !       none
;1254     1970  1 !
;1255     1971  1 ! Routine Value:
;1256     1972  1 !       none
;1257     1973  1 !
;1258     1974  1 ! Routines Called:
;1259     1975  1 !
;1260     1976  1 !
;1261     1977  1 ! Side Effects:
;1262     1978  1 !       none
;1263     1979  1 !
;1264     1980  1 !--
;1265     1981  1
;1266     1982  2     BEGIN
;1267     1983  2
;1268     1984  2     BUILTIN
;1269     1985  2         ACTUALCOUNT,
;1270     1986  2         ACTUALPARAMETER;
;1271     1987  2
;1272     1988  2     LOCAL
;1273     1989  2         RETLEN          : LONG,
;1274     1990  2         BYTES           : LONG,
;1275     1991  2         USERBUF         : LONG,
;1276     1992  2         COUNT           : BYTE,
;1277     1993  2         LENGTH          : WORD,
;1278     1994  2         ADDR            : LONG,
;1279     1995  2         STATUS          : LONG,
;1280     1996  2         FDL_FAB_PTR     : REF BLOCK [ ,BYTE ],
;1281     1997  2         FLAGS           : REF BLOCK [ ,BYTE ],
;1282     1998  2         FDL_FILE        : REF DESC_BLK,
;1283     1999  2         FDL_RESNAM      : REF DESC_BLK,
;1284     2000  2         NAME_BLOCK      : REF BLOCK [ ,BYTE ],
;1285     2001  2         TEMP_DESC       : DESC_BLK PRESET ( [ DSC$B_CLASS ] = DSC$K_CLASS_S,
;1286     2002  2                                             [ DSC$B_DTYPE ] = DSC$K_DTYPE_T );
;1287     2003  2
;1288     2004  2     ! Set up handler to control errors
;1289     2005  2     !
;1290     2006  2     LIB$ESTABLISH ( HANDLER );
;1291     2007  2
;1292     2008  2     ! Check out the arguments
;1293     2009  2     !
;1294     2010  2     COUNT = ACTUALCOUNT ();
;1295     2011  2
;1296     2012  2     ! Clear the valid stmnt flag as a fix...  should always be cleared on entry
;1297     2013  2     !
;1298     2014  2     FDL$AB_CTRL [ FDL$V_STVALID ] = _CLEAR;
;1299     2015  2
;1300     2016  3     IF ( .COUNT LSS 4 ) OR ( .COUNT GTR 8 )
;1301     2017  2     THEN
;1302     2018  2         RETURN FDL$_ILL_ARG;
;1303     2019  2
```

```
;1304    2020  2      FLAGS = ACTUALPARAMETER (1);
;1305    2021  2      FDL$AB_GENFAB = ACTUALPARAMETER (2);
;1306    2022  2      FDL$AB_GENRAB = ACTUALPARAMETER (3);
;1307    2023  2
;1308    2024  2      ! Are the FAB and RAB kosher?
;1309    2025  2      !
;1310    2026  3      IF ( .FDL$AB_GENFAB EQLU 0 ) OR ( .FDL$AB_GENRAB EQLU 0 )
;1311    2027  2      THEN
;1312    2028  2          RETURN FDL$_ILL_ARG;
;1313    2029  2
;1314    2030  3      IF ( ..FDL$AB_GENFAB EQLU 0 ) OR ( ..FDL$AB_GENRAB EQLU 0 )
;1315    2031  2      THEN
;1316    2032  2          RETURN FDL$_ILL_ARG;
;1317    2033  2
;1318    2034  2      ! Get the 'real' control blocks
;1319    2035  2      !
;1320    2036  2      FDL$AB_GENFAB = ..FDL$AB_GENFAB;
;1321    2037  2      FDL$AB_GENRAB = ..FDL$AB_GENRAB;
;1322    2038  2
;1323    2039  2      FDL_FILE = ACTUALPARAMETER (4);
;1324    2040  2
;1325    2041  2      IF .COUNT GTR 4
;1326    2042  2      THEN
;1327    2043  2          FDL_RESNAM = ACTUALPARAMETER (5)
;1328    2044  2      ELSE
;1329    2045  2          FDL_RESNAM = 0;
;1330    2046  2
;1331    2047  2      IF .COUNT GTR 5
;1332    2048  2      THEN
;1333    2049  2          FDL$AB_OUT_STRING = ACTUALPARAMETER (6)
;1334    2050  2      ELSE
;1335    2051  2          FDL$AB_OUT_STRING = 0;
;1336    2052  2
;1337    2053  2      IF .COUNT GTR 6
;1338    2054  2      THEN
;1339    2055  2          FDL$GL_INVBLK_PTR = ACTUALPARAMETER (7)
;1340    2056  2      ELSE
;1341    2057  2          FDL$GL_INVBLK_PTR = 0;
;1342    2058  2
;1343    2059  2      ! Setup to output the length of the returned string if required.
;1344    2060  2      !
;1345    2061  2      IF .COUNT GTR 7
;1346    2062  2      THEN
;1347    2063  2
;1348    2064  2          RETLEN  = ACTUALPARAMETER (8)
;1349    2065  2
;1350    2066  2      ELSE
;1351    2067  2
;1352    2068  2          RETLEN  = 0;
;1353    2069  2
;1354    2070  2      ! Are the FAB and RAB good ones?
;1355    2071  2      !
;1356    2072  2      IF .FDL$AB_GENFAB [ FAB$B_BID ] NEQU FAB$C_BID
;1357    2073  2      THEN
;1358    2074  3          BEGIN
;1359    2075  3
;1360    2076  3          IF .FDL$GL_INVBLK_PTR NEQU 0
```

```
;  1361     2077  3          THEN
;  1362     2078  3              .FDL$GL_INVBLK_PTR = .FDL$AB_GENFAB;
;  1363     2079  3
;  1364     2080  3          SIGNAL ( FDL$_INVBLK,1,.FDL$AB_GENFAB );
;  1365     2081  3
;  1366     2082  2          END;
;  1367     2083  2
;  1368     2084  2      IF .FDL$AB_GENRAB [ RAB$B_BID ] NEQU RAB$C_BID
;  1369     2085  2      THEN
;  1370     2086  3          BEGIN
;  1371     2087  3
;  1372     2088  3          IF .FDL$GL_INVBLK_PTR NEQU 0
;  1373     2089  3          THEN
;  1374     2090  3              .FDL$GL_INVBLK_PTR = .FDL$AB_GENRAB;
;  1375     2091  3
;  1376     2092  3          SIGNAL ( FDL$_INVBLK,1,.FDL$AB_GENRAB );
;  1377     2093  3
;  1378     2094  2          END;
;  1379     2095  2
;  1380     2096  2      ! Setup the flag bits
;  1381     2097  2      !
;  1382     2098  2      FDL$AB_CTRL [ FDL$V_DCL ] = _CLEAR;
;  1383     2099  2      FDL$AB_CTRL [ FDL$V_GCALL ] = _CLEAR;
;  1384     2100  2      FDL$AB_CTRL [ FDL$V_STRING_SPEC ] = _CLEAR;
;  1385     2101  2      FDL$AB_CTRL [ FDL$V_FULLGEN ] = _CLEAR;
;  1386     2102  2
;  1387     2103  2      IF .FLAGS NEQU 0
;  1388     2104  2      THEN
;  1389     2105  3          BEGIN
;  1390     2106  3
;  1391     2107  3          FDL$AB_CTRL [ FDL$V_DCL ] = .FLAGS [ FDL$V_SIGNAL ];
;  1392     2108  3          FDL$AB_CTRL [ FDL$V_GCALL ] = .FLAGS [ FDL$V_$CALLBACK ];
;  1393     2109  3          FDL$AB_CTRL [ FDL$V_STRING_SPEC ] = .FLAGS [ FDL$V_FDL_STRING ];
;  1394     2110  3          FDL$AB_CTRL [ FDL$V_FULLGEN ] = .FLAGS [ FDL$V_FULL_OUTPUT ];
;  1395     2111  3
;  1396     2112  2          END;
;  1397     2113  2
;  1398     2114  2      ! Make sure the args are self-consistent
;  1399     2115  2      !
;  1400     2116  3      IF (
;  1401     2117  4      (( .FDL$AB_CTRL [ FDL$V_STRING_SPEC ] ) AND ( .FDL$AB_OUT_STRING EQLU 0 ))
;  1402     2118  3          OR
;  1403     2119  4      (( NOT .FDL$AB_CTRL [ FDL$V_STRING_SPEC ] ) AND ( .FDL_FILE EQLU 0 ))
;  1404     2120  2      ) THEN
;  1405     2121  2          RETURN FDL$_ILL_ARG;
;  1406     2122  2
;  1407     2123  2      IF .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
;  1408     2124  2      THEN
;  1409     2125  3          RET_ON_ERROR ( LIB$ANALYZE_SDESC ( .FDL$AB_OUT_STRING,LENGTH,ADDR ) )
;  1410     2126  2      ELSE
;  1411     2127  2          RET_ON_ERROR ( LIB$ANALYZE_SDESC ( .FDL_FILE,LENGTH,ADDR ) );
;  1412     2128  2
;  1413     2129  3      IF (( NOT .FDL$AB_CTRL [ FDL$V_STRING_SPEC ] ) AND ( .ADDR EQLU 0 ))
;  1414     2130  2      THEN
;  1415     2131  2          RETURN FDL$_ILL_ARG;
;  1416     2132  2
;  1417     2133  2      IF .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
```

```
; 1418          2134  2      THEN
; 1419          2135  2          RET_ON_ERROR( LIB$SCOPY_R_DX(UPLIT(0),UPLIT(0),.FDL$AB_OUT_STRING));
; 1420          2136  2
; 1421          2137  2      ! Setup the output destination
; 1422          2138  2      !
; 1423          2139  2      IF .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
; 1424          2140  2      THEN
; 1425          2141  3          BEGIN
; 1426          2142  3
; 1427          2143  3          ! Setup the string buffer
; 1428          2144  3          !
; 1429          2145  3          BYTES = FDL$K_MAXLINE;
; 1430          2146  3
; 1431          2147  3          IF NOT LIB$GET_VM ( BYTES,USERBUF )
; 1432          2148  3          THEN
; 1433          2149  3              RETURN FDL$_INSVIRMEM;
; 1434          2150  3
; 1435          2151  3          CH$FILL ( 0,.BYTES,.USERBUF );
; 1436          2152  3          FDL$AB_LINE [ DSC$A_POINTER ] = .USERBUF;
; 1437          2153  3          FDL$AB_LINE [ DSC$W_LENGTH ] = FDL$K_MAXLINE;
; 1438          2154  3
; 1439          2155  3          END
; 1440          2156  2      ELSE
; 1441          2157  3          BEGIN
; 1442          2158  3
; 1443          2159  3          ! Set up the fab and name block
; 1444          2160  3          !
; 1445          2161  3          ! Find the number of bytes needed for all of this which is:
; 1446          2162  3          ! Space for the fab, name block, expanded and resultant string buffers
; 1447          2163  3          ! the user buffer in the rab.
; 1448          2164  3          !
; 1449          2165  3          ! The space is allocated:
; 1450          2166  3          !
; 1451          2167  3          !                 -------------------------
; 1452          2168  3          !                |          fab            | fab$k_bln
; 1453          2169  3          !                 -------------------------
; 1454          2170  3          !                |          nam            | nam$k_bln
; 1455          2171  3          !                 -------------------------
; 1456          2172  3          !                |   expanded str buffer   | esa_buf_siz
; 1457          2173  3          !                 -------------------------
; 1458          2174  3          !                |  resultant str buffer   | rsa_buf_siz
; 1459          2175  3          !                 -------------------------
; 1460          2176  3          !                |       user buffer       | fdl$k_maxline
; 1461          2177  3          !                 -------------------------
; 1462          2178  3          !
; 1463          2179  3          !
; 1464          2180  3          ! Get the address space. NOTE: Since the user never sees this buffer we
; 1465          2181  3          ! can deallocate it when we exit so use lib$get_vm
; 1466          2182  3          !
; 1467          2183  3          BYTES = FAB$K_BLN + NAM$K_BLN + ESA_BUF_SIZ + RSA_BUF_SIZ +
; 1468          2184  3                                                              FDL$K_MAXLINE;
; 1469          2185  3
; 1470          2186  3          IF NOT LIB$GET_VM( BYTES,FDL_FAB_PTR )
; 1471          2187  3          THEN
; 1472          2188  3              RETURN FDL$_INSVIRMEM;
; 1473          2189  3
; 1474          2190  3          ! Zero the space
```

```
;  1475          2191  3          !
;  1476          2192  3          CH$FILL ( 0,.BYTES,.FDL_FAB_PTR );
;  1477          2193  3
;  1478          2194  3          ! Set up the location of the user buffer
;  1479          2195  3          !
;  1480          2196  3          USERBUF = .FDL_FAB_PTR + FAB$K_BLN + NAM$K_BLN + ESA_BUF_SIZ +
;  1481          2197  3                                                           RSA_BUF_SIZ;
;  1482          2198  3
;  1483          2199  3          ! Initialize the FDL FILE fabs and rabs
;  1484          2200  3          !
;  1485        P 2201  3          $FAB_INIT ( FAB = .FDL_FAB_PTR,
;  1486        P 2202  3                      CTX = FDL$_OPENOUT,
;  1487        P 2203  3                      DNM = '.FDL',
;  1488        P 2204  3                      FAC = PUT,
;  1489        P 2205  3                      FNS = .LENGTH,
;  1490        P 2206  3                      FNA = .ADDR,
;  1491        P 2207  3                      FOP = <SQO,NAM,MXV>,
;  1492        P 2208  3                      ORG = SEQ,
;  1493        P 2209  3                      RAT = CR,
;  1494        P 2210  3                      RFM = VAR,
;  1495          2211  3                      NAM = .FDL_FAB_PTR + FAB$K_BLN );
;  1496          2212  3
;  1497        P 2213  3          $NAM_INIT ( NAM = .FDL_FAB_PTR + FAB$K_BLN,
;  1498        P 2214  3                      ESA = .FDL_FAB_PTR + FAB$K_BLN + NAM$K_BLN,
;  1499        P 2215  3                      ESS = ESA_BUF_SIZ,
;  1500        P 2216  3                      RSA = .FDL_FAB_PTR + FAB$K_BLN + NAM$K_BLN + ESA_BUF_SIZ,
;  1501          2217  3                      RSS = RSA_BUF_SIZ );
;  1502          2218  3
;  1503        P 2219  3          $RAB_INIT ( RAB = FDL$AB_FDL_RAB,
;  1504        P 2220  3                      CTX = FDL$_OPENOUT,
;  1505        P 2221  3                      FAB = .FDL_FAB_PTR,
;  1506        P 2222  3                      RAC = SEQ,
;  1507        P 2223  3                      ROP = WBH,
;  1508        P 2224  3                      RBF = .USERBUF,
;  1509        P 2225  3                      RSZ = FDL$K_MAXLINE,
;  1510        P 2226  3                      UBF = .USERBUF,
;  1511          2227  3                      USZ = FDL$K_MAXLINE );
;  1512          2228  3
;  1513          2229  3          ! Create the file
;  1514          2230  3          !
;  1515          2231  3          $CREATE( FAB=.FDL_FAB_PTR,ERR=FDL$$RMS_OPEN_ERROR );
;  1516          2232  3
;  1517          2233  3          ! Connect a stream
;  1518          2234  3          !
;  1519          2235  3          $CONNECT( RAB=FDL$AB_FDL_RAB,ERR=FDL$$RMS_OPEN_ERROR );
;  1520          2236  3
;  1521          2237  3          ! Errors from now on are write errors
;  1522          2238  3          !
;  1523          2239  3          FDL$AB_FDL_RAB [ RAB$L_CTX ] = FDL$_WRITEERR;
;  1524          2240  3
;  1525          2241  3          ! Initialize the line buffer
;  1526          2242  3          !
;  1527          2243  3          FDL$AB_LINE [ DSC$A_POINTER ] = .USERBUF;
;  1528          2244  3          FDL$AB_LINE [ DSC$W_LENGTH ] = FDL$K_MAXLINE;
;  1529          2245  3
;  1530          2246  3          ! Give the user the result name if he wants it
;  1531          2247  3          !
```

```
;  1532     2248  3          IF .FDL_RESNAM NEQ 0
;  1533     2249  3          THEN
;  1534     2250  4              BEGIN
;  1535     2251  4
;  1536     2252  4              IF ..FDL_RESNAM NEQ 0
;  1537     2253  4              THEN
;  1538     2254  5                  BEGIN
;  1539     2255  5
;  1540     2256  5                  ! Locate the name block
;  1541     2257  5                  !
;  1542     2258  5                  NAME_BLOCK = .FDL_FAB_PTR + FAB$K_BLN;
;  1543     2259  5
;  1544     2260  5                  ! If a result name desc was given, stuff it with
;  1545     2261  5                  ! the best name for file which was created (or attempted)
;  1546     2262  5                  !
;  1547     2263  5                  ! First try the resultant string
;  1548     2264  5                  !
;  1549     2265  5                  IF .NAME_BLOCK [ NAM$B_RSL ] NEQU 0
;  1550     2266  5                  THEN
;  1551     2267  6                      BEGIN
;  1552     2268  6
;  1553     2269  6                      TEMP_DESC [ DSC$W_LENGTH ] = .NAME_BLOCK [ NAM$B_RSL ];
;  1554     2270  6                      TEMP_DESC [ DSC$A_POINTER ] = .NAME_BLOCK [ NAM$[_RSA ];
;  1555     2271  6
;  1556     2272  6                      RET_ON_ERROR( LIB$SCOPY_DXDX( TEMP_DESC,.FDL_RESNAM ) );
;  1557     2273  6
;  1558     2274  6                      END
;  1559     2275  6
;  1560     2276  6                  ! Next try the expanded string
;  1561     2277  6                  !
;  1562     2278  5                  ELSE IF .NAME_BLOCK [ NAM$B_ESL ] NEQU 0
;  1563     2279  5                  THEN
;  1564     2280  6                      BEGIN
;  1565     2281  6
;  1566     2282  6                      TEMP_DESC [ DSC$W_LENGTH ] = .NAME_BLOCK [ NAM$B_ESL ];
;  1567     2283  6                      TEMP_DESC [ DSC$A_POINTER ] = .NAME_BLOCK [ NAM$[_ESA ];
;  1568     2284  6
;  1569     2285  7                      RET_ON_ERROR( LIB$SCOPY_DXDX( TEMP_DESC,.FDL_RESNAM ) )
;  1570     2286  7
;  1571     2287  6                      END
;  1572     2288  6
;  1573     2289  6                  ! If all else fails use the name string
;  1574     2290  6                  !
;  1575     2291  5                  ELSE
;  1576     2292  6                      BEGIN
;  1577     2293  6
;  1578     2294  6                      TEMP_DESC [ DSC$W_LENGTH ] = .FDL_FAB_PTR [ FAB$B_FNS ];
;  1579     2295  6                      TEMP_DESC [ DSC$A_POINTER ] = .FDL_FAB_PTR [ FAB$[_FNA ];
;  1580     2296  6
;  1581     2297  7                      RET_ON_ERROR( LIB$SCOPY_DXDX( TEMP_DESC,.FDL_RESNAM ) )
;  1582     2298  7
;  1583     2299  5                      END;
;  1584     2300  5
;  1585     2301  5                  ! Return the length if requested.
;  1586     2302  5                  !
;  1587     2303  5                  IF .RETLEN NEQ 0
;  1588     2304  5                  THEN
```

```
;  1589     2305  6                    BEGIN
;  1590     2306  6
;  1591     2307  6                    .RETLEN     = .TEMP_DESC [ DSC$W_LENGTH ];
;  1592     2308  6
;  1593     2309  5                    END;
;  1594     2310  4                END;
;  1595     2311  3            END;
;  1596     2312  2        END;
;  1597     2313  2
;  1598     2314  2        ! Set up the control block blk
;  1599     2315  2        !
;  1600     2316  2        FDL$AB_BLOCK_BLK [ FDL$C_FAB ] = .FDL$AB_GENFAB;
;  1601     2317  2        FDL$AB_BLOCK_BLK [ FDL$C_RAB ] = .FDL$AB_GENRAB;
;  1602     2318  2        FDL$AB_BLOCK_BLK [ FDL$C_NAM ] = .FDL$AB_GENFAB [ FAB$L_NAM ];
;  1603     2319  2
;  1604     2320  2        ! Now go generate the FDL SPEC
;  1605     2321  2        !
;  1606     2322  2        STATUS = FDL$$GEN_SPEC ();
;  1607     2323  2
;  1608     2324  2        ! Return the length if requested.
;  1609     2325  2        !
;  1610     2326  2        IF .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
;  1611     2327  2        THEN
;  1612     2328  3            BEGIN
;  1613     2329  3
;  1614     2330  3            IF .RETLEN NEQ 0
;  1615     2331  3            THEN
;  1616     2332  4                BEGIN
;  1617     2333  4
;  1618     2334  4                LIB$ANALYZE_SDESC ( .FDL$AB_OUT_STRING,LENGTH,ADDR );
;  1619     2335  4                .RETLEN = .LENGTH;
;  1620     2336  4
;  1621     2337  3                END;
;  1622     2338  2            END;
;  1623     2339  2
;  1624     2340  2        IF NOT .FDL$AB_CTRL [ FDL$V_STRING_SPEC ]
;  1625     2341  2        THEN
;  1626     2342  3            BEGIN
;  1627     2343  3
;  1628     2344  3            ! Close down the FDL file and exit
;  1629     2345  3            !
;  1630     2346  3            $DISCONNECT( RAB=FDL$AB_FDL_RAB );
;  1631     2347  3
;  1632     2348  3            $CLOSE( FAB=.FDL_FAB_PTR );
;  1633     2349  3
;  1634     2350  3                ! Deallocate the memory for the fab etc
;  1635     2351  3                !
;  1636     2352  4                BEGIN
;  1637     2353  4                LOCAL STATUS;
;  1638     2354  4
;  1639     2355  5                IF NOT ( STATUS = LIB$FREE_VM( BYTES,FDL_FAB_PTR ))
;  1640     2356  4                THEN
;  1641     2357  4                    SIGNAL_STOP ( .STATUS );
;  1642     2358  3                END;
;  1643     2359  3
;  1644     2360  2            END;
;  1645     2361  2
```

```
;  1646          2362  2     RETURN .STATUS;
;  1647          2363  2
;  1648          2364  1     END;


                                                .PSECT  _FDL$PLIT,NOWRT,NOEXE,  SHR,  PIC,2

                                    00000000  00004 P.AAB:  .LONG   0                                                    ;
                                    00000000  00008 P.AAC:  .LONG   0                                                    ;
                                 4C  44  46  2E  0000C P.AAD:  .ASCII  \.FDL\                                            ;

                                                $RMS_PTR=           FDL$AB_FDL_RAB

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

                                        0FFC 00000          .ENTRY  FDL$GENERATE, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  1907
                                                                    R10,R11                                             ;
                                 5B 00000000G   00   9E 00002       MOVAB   FDL$AB_CTRL, R11                            ;
                                 5A 00000000'   00   9E 00009       MOVAB   FDL$AB_GENRAB, R10                          ;
                                 5E             1C   C2 00010       SUBL2   #28, SP                                     ;
                              14 AE 010E0000    8F   D0 00013       MOVL    #17694720, TEMP_DESC                        ;  2002
                                        18 AE   D4 0001B            CLRL    TEMP_DESC+4                                 ;
                                 00000000V      00   9F 0001E       PUSHAB  HANDLER                                     ;  2006
                           00000000G  00        01   FB 00024       CALLS   #1, LIB$ESTABLISH                           ;
                                 51             6C   90 0002B       MOVB    (AP), COUNT                                 ;  2010
                              02 AB             04   8A 0002E       BICB2   #4, FDL$AB_CTRL+2                           ;  2014
                                 04             51   91 00032       CMPB    COUNT, #4                                   ;  2016
                                                03   1E 00035       BGEQU   2$                                          ;
                                             013A   31 00037 1$:     BRW     20$                                         ;
                                 08             51   91 0003A 2$:    CMPB    COUNT, #8                                   ;
                                                F8   1A 0003D       BGTRU   1$                                          ;
                                 52       04    AC   D0 0003F       MOVL    4(AP), FLAGS                                ;  2020
                              FC AA       08    AC   7D 00043       MOVQ    8(AP), FDL$AB_GENFAB                        ;  2021
                                 50       FC AA   D0 00048         MOVL    FDL$AB_GENFAB, R0                           ;  2026
                                                E9   13 0004C       BEQL    1$                                          ;
                                                6A   D5 0004E       TSTL    FDL$AB_GENRAB                               ;
                                                E5   13 00050       BEQL    1$                                          ;
                                                60   D5 00052       TSTL    (R0)                                        ;  2030
                                                E1   13 00054       BEQL    1$                                          ;
                                 53             6A   D0 00056       MOVL    FDL$AB_GENRAB, R3                           ;
                                                63   D5 00059       TSTL    (R3)                                        ;
                                                DA   13 0005B       BEQL    1$                                          ;
                              FC AA             60   D0 0005D       MOVL    (R0), FDL$AB_GENFAB                         ;  2036
                                 50             6A   D0 00061       MOVL    FDL$AB_GENRAB, R0                           ;  2037
                                 6A             60   D0 00064       MOVL    (R0), FDL$AB_GENRAB                         ;
                                 53       10    AC   D0 00067       MOVL    16(AP), FDL_FILE                            ;  2039
                                 04             51   91 0006B       CMPB    COUNT, #4                                   ;  2041
                                                06   1B 0006E       BLEQU   3$                                          ;
                                 58       14    AC   D0 00070       MOVL    20(AP), FDL_RESNAM                          ;  2043
                                                02   11 00074       BRB     4$                                          ;
                                                58   D4 00076 3$:    CLRL    FDL_RESNAM                                  ;  2045
                                 05             51   91 00078 4$:    CMPB    COUNT, #5                                   ;  2047
                                                0A   1B 0007B       BLEQU   5$                                          ;
                           00000000G  00        18   AC   D0 0007D  MOVL    24(AP), FDL$AB_OUT_STRING                   ;  2049
                                                06   11 00085       BRB     6$                                          ;
```

```
                                   00000000G  00  D4 00087 5$:      CLRL    FDL$AB_OUT_STRING                              : 2051
                              06              51  91 0008D 6$:      CMPB    COUNT, #6                                      : 2053
                                              0A  1B 00090          BLEQU   7$                                             :
                    00000000G  00         1C  AC  D0 00092          MOVL    28(AP), FDL$GL_INVBLK_PTR                      : 2055
                                              06  11 0009A          BRB     8$                                             :
                                   00000000G  00  D4 0009C 7$:      CLRL    FDL$GL_INVBLK_PTR                              : 2057
                              07              51  91 000A2 8$:      CMPB    COUNT, #7                                      : 2061
                                              06  1B 000A5          BLEQU   9$                                             :
                              59          20  AC  D0 000A7          MOVL    32(AP), RETLEN                                 : 2064
                                              02  11 000AB          BRB     10$                                            :
                                              59  D4 000AD 9$:      CLRL    RETLEN                                         : 2068
                              51          FC  AA  D0 000AF 10$:     MOVL    FDL$AB_GENFAB, R1                              : 2072
                              03              61  91 000B3          CMPB    (R1), #3                                       :
                                              1D  13 000B6          BEQL    12$                                            :
                              50 00000000G    00  D0 000B8          MOVL    FDL$GL_INVBLK_PTR, R0                          : 2076
                                              03  13 000BF          BEQL    11$                                            :
                              60              51  D0 000C1          MOVL    R1, (R0)                                       : 2078
                                              51  DD 000C4 11$:     PUSHL   R1                                             : 2080
                                              01  DD 000C6          PUSHL   #1                                             :
                                   00000000G  8F  DD 000C8          PUSHL   #FDL$_INVBLK                                   :
                    00000000G  00             03  FB 000CE          CALLS   #3, LIB$SIGNAL                                 :
                              51              6A  D0 000D5 12$:     MOVL    FDL$AB_GENRAB, R1                              : 2084
                              01              61  91 000D8          CMPB    (R1), #1                                       :
                                              1D  13 000DB          BEQL    14$                                            :
                              50 00000000G    00  D0 000DD          MOVL    FDL$GL_INVBLK_PTR, R0                          : 2088
                                              03  13 000E4          BEQL    13$                                            :
                              60              51  D0 000E6          MOVL    R1, (R0)                                       : 2090
                                              51  DD 000E9 13$:     PUSHL   R1                                             : 2092
                                              01  DD 000EB          PUSHL   #1                                             :
                                   00000000G  8F  DD 000ED          PUSHL   #FDL$_INVBLK                                   :
                    00000000G  00             03  FB 000F3          CALLS   #3, LIB$SIGNAL                                 :
                           01  AB       1818  8F  AA 000FA 14$:     BICW2   #6168, FDL$AB_CTRL+1                           : 2101
                                              52  D5 00100          TSTL    FLAGS                                          : 2103
                                              27  13 00102          BEQL    15$                                            :
  01  AB        01            03              62  F0 00104          INSV    (FLAGS), #3, #1, FDL$AB_CTRL+1                 : 2107
      50        62            01              04  EF 0010A          EXTZV   #4, #1, (FLAGS), R0                            : 2108
  02  AB        01            03              50  F0 0010F          INSV    R0, #3, #1, FDL$AB_CTRL+2                      :
      50        62            01              01  EF 00115          EXTZV   #1, #1, (FLAGS), R0                            : 2109
  01  AB        01            04              50  F0 0011A          INSV    R0, #4, #1, FDL$AB_CTRL+1                      :
      50        62            01              03  EF 00120          EXTZV   #3, #1, (FLAGS), R0                            : 2110
  02  AB        01            04              50  F0 00125          INSV    R0, #4, #1, FDL$AB_CTRL+2                      :
      50    01  AB            01              04  EF 0012B 15$:     EXTZV   #4, #1, FDL$AB_CTRL+1, R0                      : 2117
                              0B              50  E9 00131          BLBC    R0, 16$                                        :
                                   00000000G  00  D5 00134          TSTL    FDL$AB_OUT_STRING                              :
                                              38  13 0013A          BEQL    20$                                            :
                              07              50  E8 0013C          BLBS    R0, 17$                                        : 2119
                                              53  D5 0013F 16$:     TSTL    FDL_FILE                                       :
                                              31  13 00141          BEQL    20$                                            :
                              0E              50  E9 00143          BLBC    R0, 18$                                        : 2123
                                          04  AE  9F 00146 17$:     PUSHAB  ADDR                                           : 2125
                                          0C  AE  9F 00149          PUSHAB  LENGTH                                         :
                                   00000000G  00  DD 0014C          PUSHL   FDL$AB_OUT_STRING                              :
                                              08  11 00152          BRB     19$                                            :
                                          04  AE  9F 00154 18$:     PUSHAB  ADDR                                           : 2127
                                          0C  AE  9F 00157          PUSHAB  LENGTH                                         :
                                              53  DD 0015A          PUSHL   FDL_FILE                                       :
                    00000000G  00             03  FB 0015C 19$:     CALLS   #3, LIB$ANALYZE_SDESC                          :
```

```
                              32               50 E9 00163         BLBC    STATUS, 23$                              ;
     50       01   AB         01               04 EF 00166         EXTZV   #4, #1, FDL$AB_CTRL+1, R0                ; 2129
                              10               50 E8 0016C         BLBS    R0, 22$                                  ;
                                         04    AE D5 0016F         TSTL    ADDR                                     ;
                                               08 12 00172         BNEQ    21$                                      ;
                              50 00000000G     8F D0 00174 20$:    MOVL    #FDL$_ILL_ARG, R0                        ; 2131
                                                  04 0017B         RET                                              ;
                              1D               50 E9 0017C 21$:    BLBC    R0, 24$                                  ; 2133
                                 00000000G     00 DD 0017F 22$:    PUSHL   FDL$AB_OUT_STRING                        ; 2135
                                 00000000'     00 9F 00185         PUSHAB  P.AAC                                    ;
                                 00000000'     00 9F 0018B         PUSHAB  P.AAB                                    ;
                    00000000G 00               03 FB 00191         CALLS   #3, LIB$SCOPY_R_DX                       ;
                              01               50 E8 00198 23$:    BLBS    STATUS, 24$                              ;
                                                  04 0019B         RET                                              ;
                    30     01 AB               04 E1 0019C 24$:    BBC     #4, FDL$AB_CTRL+1, 25$                   ; 2139
                           10 AE       0400    8F 3C 001A1         MOVZWL  #1024, BYTES                             ; 2145
                                               5E DD 001A7         PUSHL   SP                                       ; 2147
                                         14    AE 9F 001A9         PUSHAB  BYTES                                    ;
                    00000000G 00               02 FB 001AC         CALLS   #2, LIB$GET_VM                           ;
                              31               50 E9 001B3         BLBC    R0, 26$                                  ;
   10   AE        00          6E               00 2C 001B6         MOVC5   #0, (SP), #0, BYTES, @USERBUF            ; 2151
                                         00    BE    001BC                                                          ;
                    00000000G 00               6E D0 001BE         MOVL    USERBUF, FDL$AB_LINE+4                   ; 2152
                    00000000G 00       0400    8F B0 001C5         MOVW    #1024, FDL$AB_LINE                       ; 2153
                                       0152    31    001CE         BRW     32$                                      ; 2139
                           10 AE       06AE    8F 3C 001D1 25$:    MOVZWL  #1710, BYTES                             ; 2183
                                         0C    AE 9F 001D7         PUSHAB  FDL_FAB_PTR                              ; 2186
                                         14    AE 9F 001DA         PUSHAB  BYTES                                    ;
                    00000000G 00               02 FB 001DD         CALLS   #2, LIB$GET_VM                           ;
                              08               50 E8 001E4         BLBS    R0, 27$                                  ;
                              50 00000000G     8F D0 001E7 26$:    MOVL    #FDL$_INSVIRMEM, R0                      ; 2188
                                                  04 001EE         RET                                              ;
                              56         0C    AE D0 001EF 27$:    MOVL    FDL_FAB_PTR, R6                          ; 2192
   10   AE        00          6E               00 2C 001F3         MOVC5   #0, (SP), #0, BYTES, (R6)                ;
                                               66    001F9                                                          ;
                              6E       02AE    C6 9E 001FA         MOVAB   686(R6), USERBUF                         ; 2196
 0050   8F        00          6E               00 2C 001FF         MOVC5   #0, (SP), #0, #80, (R6)                  ; 2211
                                               66    00206                                                          ;
                              66       5003    8F B0 00207         MOVW    #20483, (R6)                             ;
                           04 A6 01000042      8F D0 0020C         MOVL    #16777282, 4(R6)                         ;
                           16 A6               01 90 00214         MOVB    #1, 22(R6)                               ;
                           18 A6 00000000G     8F D0 00218         MOVL    #FDL$_OPENOUT, 24(R6)                    ;
                           1D A6       0200    8F B0 00220         MOVW    #512, 29(R6)                             ;
                           1F A6               02 90 00226         MOVB    #2, 31(R6)                               ;
                              57         50    A6 9E 0022A         MOVAB   80(R6), R7                               ;
                           28 A6               57 D0 0022E         MOVL    R7, 40(R6)                               ;
                           2C A6         04    AE D0 00232         MOVL    ADDR, 44(R6)                             ;
                           30 A6 00000000'     00 9E 00237         MOVAB   P.AAD, 48(R6)                            ;
                           34 A6         08    AE 90 0023F         MOVB    LENGTH, 52(R6)                           ;
                           35 A6               04 90 00244         MOVB    #4, 53(R6)                               ;
 0060   8F        00          6E               00 2C 00248         MOVC5   #0, (SP), #0, #96, (R7)                  ; 2217
                                               67    0024F                                                          ;
                              67       6002    8F B0 00250         MOVW    #24578, (R7)                             ;
                           02 A7               01 8E 00255         MNEGB   #1, 2(R7)                                ;
                           04 A7       01AF    C6 9E 00259         MOVAB   431(R6), 4(R7)                           ;
                           0A A7               01 8E 0025F         MNEGB   #1, 10(R7)                               ;
                           0C A7       00B0    C6 9E 00263         MOVAB   176(R6), 12(R7)                          ;
```

```
0044  8F  00  6E  00 2C 00269        MOVC5   #0, (SP), #0, #68, $RMS_PTR                    ; 2227
                  B8  AA 00270
                B8  AA    3401  8F  B0 00272        MOVW    #17409, $RMS_PTR
                BC  AA    0400  8F  3C 00278        MOVZWL  #1024, $RMS_PTR+4
                D0  AA 00000000G  8F  D0 0027E      MOVL    #FDL$_OPENOUT, $RMS_PTR+24
                      D6  AA  94 00286              CLRB    $RMS_PTR+30
                D8  AA 04000400   8F  D0 00289      MOVL    #67109888, $RMS_PTR+32
                DC  AA            6E  D0 00291      MOVL    USERBUF, $RMS_PTR+36
                E0  AA            6E  D0 00295      MOVL    USERBUF, $RMS_PTR+40
                F4  AA            56  D0 00299      MOVL    R6, $RMS_PTR+60
                    00000000G     00  9F 0029D      PUSHAB  FDL$$RMS_OPEN_ERROR                 ; 2231
                                  56  DD 002A3      PUSHL   R6
          00000000G  00           02  FB 002A5      CALLS   #2, SYS$CREATE
                    00000000G     00  9F 002AC      PUSHAB  FDL$$RMS_OPEN_ERROR                 ; 2235
                          B8      AA  9F 002B2      PUSHAB  FDL$AB_FDL_RAB
          00000000G  00           02  FB 002B5      CALLS   #2, SYS$CONNECT
                D0  AA 00000000G  8F  D0 002BC      MOVL    #FDL$_WRITEERR, FDL$AB_FDL_RAB+24   ; 2239
          00000000G  00           6E  D0 002C4      MOVL    USERBUF, FDL$AB_LINE+4              ; 2243
          00000000G  00     0400  8F  B0 002CB      MOVW    #1024, FDL$AB_LINE                  ; 2244
                                  58  D5 002D4      TSTL    FDL_RESNAM                          ; 2248
                                  4B  13 002D6      BEQL    32$
                                  68  D5 002D8      TSTL    (FDL_RESNAM)                        ; 2252
                                  47  13 002DA      BEQL    32$
                              50  57  D0 002DC      MOVL    R7, NAME_BLOCK                      ; 2258
                                03  A0  95 002DF      TSTB    3(NAME_BLOCK)                       ; 2265
                                  0C  13 002E2      BEQL    28$
                          14  AE  03  A0  9B 002E4      MOVZBW  3(NAME_BLOCK), TEMP_DESC        ; 2269
                          18  AE  04  A0  D0 002E9      MOVL    4(NAME_BLOCK), TEMP_DESC+4      ; 2270
                                  1B  11 002EE      BRB     30$                                 ; 2272
                                0B  A0  95 002F0  28$:  TSTB    11(NAME_BLOCK)                      ; 2278
                                  0C  13 002F3      BEQL    29$
                          14  AE  0B  A0  9B 002F5      MOVZBW  11(NAME_BLOCK), TEMP_DESC       ; 2282
                          18  AE  0C  A0  D0 002FA      MOVL    12(NAME_BLOCK), TEMP_DESC+4     ; 2283
                                  0A  11 002FF      BRB     30$                                 ; 2285
                          14  AE  34  A6  9B 00301  29$:  MOVZBW  52(R6), TEMP_DESC           ; 2294
                          18  AE  2C  A6  D0 00306      MOVL    44(R6), TEMP_DESC+4             ; 2295
                                  58  DD 0030B  30$:  PUSHL   FDL_RESNAM                      ; 2297
                                18  AE  9F 0030D      PUSHAB  TEMP_DESC
          00000000G  00           02  FB 00310      CALLS   #2, LIB$SCOPY_DXDX
                     01           50  E8 00317      BLBS    STATUS, 31$
                                  04  00 0031A      RET
                                  59  D5 0031B  31$:  TSTL    RETLEN                          ; 2303
                                  04  13 0031D      BEQL    32$
                              69  14  AE  3C 0031F      MOVZWL  TEMP_DESC, (RETLEN)             ; 2307
                              50  FC  AA  D0 00323  32$:  MOVL    FDL$AB_GENFAB, R0           ; 2316
          00000000G  00           50  D0 00327      MOVL    R0, FDL$AB_BLOCK_BLK
          00000000G  00           6A  D0 0032E      MOVL    FDL$AB_GENRAB, FDL$AB_BLOCK_BLK+4   ; 2317
          00000000G  00     28    A0  D0 00335      MOVL    40(R0), FDL$AB_BLOCK_BLK+12         ; 2318
          00000000G  00           00  FB 0033D      CALLS   #0, FDL$$GEN_SPEC                   ; 2322
                     52           50  D0 00344      MOVL    R0, STATUS
20          01  AB                04  E1 00347      BBC     #4, FDL$AB_CTRL+1, 34$              ; 2326
                                  59  D5 0034C      TSTL    RETLEN                              ; 2330
                                  17  13 0034E      BEQL    33$
                              04  AE  9F 00350      PUSHAB  ADDR                                ; 2334
                              0C  AE  9F 00353      PUSHAB  LENGTH
                    00000000G     00  DD 00356      PUSHL   FDL$AB_OUT_STRING
          00000000G  00           03  FB 0035C      CALLS   #3, LIB$ANALYZF_SDESC
```

```
                                69      08    AE  3C 00363           MOVZWL  LENGTH, (RETLEN)                        ; 2335
                             2D    01   AB      04  E0 00367 33$:    BBS     #4, FDL$AB_LTR_+1, 35$                  ; 2340
                                       B8  AA    9F 0036C 34$:       PUSHAB  FDL$AB_FDL_RAB                          ; 2346
                          00000000G    00      01  FB 0036F         CALLS   #1, SYS$DISCONNECT                      ;
                                         0C    AE  DD 00376         PUSHL   FDL_FAB_PTR                             ; 2348
                          00000000G    00      01  FB 00379         CALLS   #1, SYS$CLOSE                           ;
                                         0C    AE  9F 00380         PUSHAB  FDL_FAB_PTR                             ; 2355
                                         14    AE  9F 00383         PUSHAB  BYTES                                   ;
                          00000000G    00      02  FB 00386         CALLS   #2, LIB$FREE_VM                         ;
                                       09        50  E8 0038D         BLBS    STATUS, 35$                             ;
                                                 50  DD 00390         PUSHL   STATUS                                  ; 2357
                          00000000G    00      01  FB 00392         CALLS   #1, LIB$STOP                            ;
                                       50        52  D0 00399 35$:    MOVL    STATUS, R0                              ; 2362
                                                 04  0039C           RET                                             ; 2364

; Routine Size:  925 bytes,    Routine Base:  _FDL$CODE + 0638
```

```
;1650    2365  1 %SBTTL  'HANDLER'
;1651    2366  1 ROUTINE HANDLER ( SIGNAL_VECTOR : REF BLOCK [ ,BYTE ],
;1652    2367  1                   MECH_VECTOR : REF BLOCK [ ,BYTE ]) =
;1653    2368  1 !++
;1654    2369  1 !
;1655    2370  1 ! Functional Description:
;1656    2371  1 !
;1657    2372  1 !       Condition handler for the fdl parser
;1658    2373  1 !
;1659    2374  1 ! Calling Sequence:
;1660    2375  1 !
;1661    2376  1 ! Input Parameters:
;1662    2377  1 !
;1663    2378  1 ! Implicit Inputs:
;1664    2379  1 !       none
;1665    2380  1 !
;1666    2381  1 ! Output Parameters:
;1667    2382  1 !
;1668    2383  1 ! Implicit Outputs:
;1669    2384  1 !       none
;1670    2385  1 !
;1671    2386  1 ! Routine Value:
;1672    2387  1 !       none
;1673    2388  1 !
;1674    2389  1 ! Side Effects:
;1675    2390  1 !       none
;1676    2391  1 !
;1677    2392  1 !--
;1678    2393  1
;1679    2394  2     BEGIN
;1680    2395  2
;1681    2396  2     LOCAL
;1682    2397  2         CONDITION_CODE  : BLOCK [ 4,BYTE ];
;1683    2398  2
;1684    2399  2     ! Get the condition code
;1685    2400  2     !
;1686    2401  2     CONDITION_CODE = .SIGNAL_VECTOR [ CHF$L_SIG_NAME ];
;1687    2402  2
;1688    2403  2     ! If an unwind is in progress return
;1689    2404  2     !
;1690    2405  3     IF ( .CONDITION_CODE EQLU SS$_UNWIND )
;1691    2406  2     THEN
;1692    2407  2         RETURN 0;
;1693    2408  2
;1694    2409  2     ! If this is not a warning skip it otherwise stuff the status
;1695    2410  2     ! with the greater of the errors
;1696    2411  2     !
;1697    2412  2     IF .CONDITION_CODE [ STS$V_SEVERITY ] NEQU STS$K_INFO
;1698    2413  2     THEN
;1699    2414  2
;1700    2415  2         ! If the current condition is ok OR this worse then anything weve seen
;1701    2416  2         ! then make this the new error
;1702    2417  2         !
;1703    2418  2         IF .FDL$AB_CTRL [ FDL$V_STATUS ] OR
;1704    2419  3                 ( .CONDITION_CODE [ STS$V_SEVERITY ] GTRU
;1705    2420  3                                         .FDL$AB_CTRL [ FDL$V_STATUS ] )
;1706    2421  2         THEN
```

```
;1707     2422  2              FDL$AB_CTRL [ FDL$V_STATUS ] = .CONDITION_CODE [ STS$V_SEVERITY ];
;1708     2423  2
;1709     2424  2      IF NOT .FDL$AB_CTRL [ FDL$V_DCL ]
;1710     2425  2      THEN
;1711     2426  2          LIB$SIG_TO_RET (
;1712     2427  2                          SIGNAL_VECTOR [ CHF$L_SIG_ARGS ],
;1713     2428  2                          MECH_VECTOR [ CHF$L_MCH_ARGS ]
;1714     2429  2                          );
;1715     2430  2
;1716     2431  2      RETURN SS$_RESIGNAL
;1717     2432  2
;1718     2433  1      END;


                                                000C 00000 HANDLER:.WORD   Save R2,R3                                      ;2366
                                  53 00000000G   00   9E 00002          MOVAB   FDL$AB_CTRL, R3
                                  52         04   AC   D0 00009          MOVL    SIGNAL_VECTOR, R2                         ;2401
                                  51         04   A2   D0 0000D          MOVL    4(R2), CONDITION_CODE
                      00000920    8F              51   D1 00011          CMPL    CONDITION_CODE, #2336                     ;2405
                                                  32   13 00018          BEQL    4$
          03              51              03      00   ED 0001A          CMPZV   #0, #3, CONDITION_CODE, #3                ;2412
                                                  14   13 0001F          BEQL    2$
          50              63              03      00   EF 00021          EXTZV   #0, #3, FDL$AB_CTRL, R0                   ;2418
                                          07      50   E8 00026          BLBS    R0, 1$
          50              51              03      00   ED 00029          CMPZV   #0, #3, CONDITION_CODE, R0                ;2420
                                                  05   1B 0002E          BLEQU   2$
          63              03              00      51   F0 00030 1$:      INSV    CONDITION_CODE, #0, #3, FDL$AB_CTRL       ;2422
                          0C      01      A3      03   E0 00035 2$:      BBS     #3, FDL$AB_CTRL+1, 3$                     ;2424
                                          08   AC   DD 0003A             PUSHL   MECH_VECTOR                               ;2428
                                                  52   DD 0003D          PUSHL   R2
                      00000000G   00              02   FB 0003F          CALLS   #2, LIB$SIG_TO_RET
                                  50      0918    8F   3C 00046 3$:      MOVZWL  #2328, R0                                 ;2431
                                                  04   00048            RET
                                                  50   D4 0004C 4$:      CLRL    R0                                        ;2433
                                                  04   0004E            RET

; Routine Size:  79 bytes,    Routine Base:  _FDL$CODE + 09D5

;1719     2434  1
;1720     2435  0 END     ELUDOM


                                                                         .EXTRN  LIB$SIGNAL, LIB$STOP

                                    PSECT SUMMARY

        Name                            Bytes                           Attributes

  _FDL$GLOBAL                             76  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
  _FDL$OWN                                 8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
  _FDL$CODE                             2596  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
```

```
;  _FDL$PLIT                    16  NOVEC,NOWRT,  RD ,NOEXE,  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       120         1       581         00:00.9


;                                      COMMAND QUALIFIERS

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

; Size:          2596 code + 100 data bytes
; Run Time:         00:58.2
; Elapsed Time:     03:00.8
; Lines/CPU Min:     2510
; Lexemes/CPU-Min: 28086
; Memory Used:  369 pages
; Compilation Complete
```

0176 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

FDLDATA
LIS

FDLUTIL
REQ

FDLSHR
MAP

FDLPARDEF
SDL

FDLCALL
LIS

CREATEFDL
LIS

FDLDRIVER
LIS

FDLGEN
LIS