```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    FFFFFFFFFFFFFFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEEEEEEEEEEE     DDD         DDD  FFFFFFFFFFFF
EEEEEEEEEEEE     DDD         DDD  FFFFFFFFFFFF
EEEEEEEEEEEE     DDD         DDD  FFFFFFFFFFFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEE              DDD         DDD  FFF
EEEEEEEEEEEEEEE  DDDDDDDDDDD     FFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    FFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    FFF
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    FFF
```

```
Va
--
00
00
00
00
00
00
00
00
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
7F
```

```
EEEEEEEEEE  DDDDDDDD    FFFFFFFFFF  EEEEEEEEEE  XX      XX  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR    NN      NN
EEEEEEEEEE  DDDDDDDD    FFFFFFFFFF  EEEEEEEEEE  XX      XX  TTTTTTTTTT  EEEEEEEEEE  RRRRRRRR    NN      NN
EE          DD      DD  FF          EE          XX      XX      TT      EE          RR      RR  NN      NN
EE          DD      DD  FF          EE          XX      XX      TT      EE          RR      RR  NN      NN
EE          DD      DD  FF          EE            XX  XX        TT      EE          RR      RR  NNNN    NN
EE          DD      DD  FF          EE            XX  XX        TT      EE          RR      RR  NNNN    NN
EEEEEEEE    DD      DD  FFFFFFFF    EEEEEEEE        XX          TT      EEEEEEEE    RRRRRRRR    NN  NN  NN
EEEEEEEE    DD      DD  FFFFFFFF    EEEEEEEE        XX          TT      EEEEEEEE    RRRRRRRR    NN  NN  NN
EE          DD      DD  FF          EE            XX  XX        TT      EE          RR  RR      NN    NNNN
EE          DD      DD  FF          EE            XX  XX        TT      EE          RR  RR      NN    NNNN
EE          DD      DD  FF          EE          XX      XX      TT      EE          RR    RR    NN      NN      ....
EE          DD      DD  FF          EE          XX      XX      TT      EE          RR    RR    NN      NN      ....
EEEEEEEEEE  DDDDDDDD    FF          EEEEEEEEEE  XX      XX      TT      EEEEEEEEEE  RR      RR  NN      NN      ....
EEEEEEEEEE  DDDDDDDD    FF          EEEEEEEEEE  XX      XX      TT      EEEEEEEEEE  RR      RR  NN      NN      ....

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

```
0001            [ IDENT ('V04-000'),
0002        { ++
0003         ******************************************************************************
0004         *                                                                            *
0005         *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                   *
0006         *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
0007         *  ALL RIGHTS RESERVED.                                                      *
0008         *                                                                            *
0009         *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0010         *  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE     *
0011         *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0012         *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0013         *  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0014         *  TRANSFERRED.                                                              *
0015         *                                                                            *
0016         *  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0017         *  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
0018         *  CORPORATION.                                                              *
0019         *                                                                            *
0020         *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0021         *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
0022         *                                                                            *
0023         *                                                                            *
0024         ******************************************************************************
0025
0026
0027
0028
0029
0030            FACILITY:       VAX/VMS EDF (EDIT/FDL) UTILITY
0031
0032            ABSTRACT:       This facility is used to create, modify, and optimize
0033                            FDL specification files.
0034
0035            ENVIRONMENT:    NATIVE/USER MODE
0036
0037            AUTHOR:         Ken F. Henderson Jr.
0038
0039            CREATION DATE:  27-Mar-1981
0040
0041            MODIFIED BY:
0042
0043                            V03-008 KFH0008         Ken Henderson          8 Aug 1983
0044                                    Changes for seperate compilation.
0045
0046                            V03-007 KFH0007         Ken Henderson         14 Apr 1983
0047                                    Changed EDF$TERM_SETUP function
0048                                    value from BYTE to INTEGER.
0049
0050                            V03-006 KFH0006         Ken Henderson         22 Nov 1982
0051                                    Removed obsolete SYS$routine
0052                                    definitions. (now from starlet.pen)
0053
0054                            V03-005 KFH0005         Ken Henderson         11 Oct 1982
0055                                    Added support for Access, ACL,
0056                                    Connect, Journal, Sharing primaries.
0057                                    Changed FDL$$PARSE to FDL$PARSE.
```

```
0058                            Modified FDL$CREATE.
0059
0060                    V03-004 KFH0004         Ken Henderson           22 Sept 1982
0061                            Added definition of LIB$WAIT;
0062                            Removed definition of FOR$SECNDS.
0063
0064                    V03-003 KFH0003         Ken Henderson           11 August 1982
0065                            Removed reference to CLI$END_PARSE.
0066
0067                    V03-002 KFH0002         Ken Henderson           2 April 1982
0068                            Added ISECNUM_PTR to call to
0069                            FDL$GLOBAL.
0070
0071                    V03-001 KFH0001         Ken Henderson           23-Mar-1982
0072                            Modified LIB$OUTPUT_HELP to fix
0073                            FT2 QAR 381
0074
0075            -- }
```

```
0077            ENVIRONMENT ('LIB$:EDFEXTERN'),
0078
0079            INHERIT (
0080
0081            'SYS$LIBRARY:STARLET',
0082            'SHRLIB$:FDLPARDEF',
0083            'LIB$:EDFSTRUCT',
0084            'LIB$:EDFCONST',
0085            'LIB$:EDFTYPE',
0086            'LIB$:EDFVAR'
0087
0088            )]
0089
0090            MODULE EDFEXTERN;
```

```
0092            { ++
0093            EXTERNAL ROUTINE DECLARATIONS:
0094
0095            The following functions and procedures are part of the VMS RunTimeLibrary.
0096            They are documented in the RTL Reference Manual and the Utilities Reference
0097            Manual.
0098            -- }
0099
0100            { +
0101            These are the 'Terminal Independent Screen Procedures'.
0102            - }
0103
0104            [ASYNCHRONOUS] FUNCTION LIB$PUT_LINE (
0105
0106                    %DESCR      TEXTX       : [VOLATILE] VARYING [A] OF CHAR;
0107                            VAR LINE_ADV    : [VOLATILE] INTEGER;
0108                                FLAGS       : [VOLATILE] INTEGER
0109
0110                                ) : INTEGER; EXTERNAL;
0111
0112            FUNCTION LIB$ERASE_LINE (
0113
0114                            VAR LINE,
0115                                COLUMN      : INTEGER
0116
0117                                ) : INTEGER; EXTERN;
0118
0119            FUNCTION LIB$ERASE_PAGE (
0120
0121                            VAR LINE,
0122                                COLUMN      : [VOLATILE] INTEGER
0123
0124                                ) : INTEGER; EXTERN;
0125
0126            FUNCTION LIB$SET_CURSOR (
0127
0128                            VAR LINE,
0129                                COLUMN      : [VOLATILE] INTEGER
0130
0131                                ) : INTEGER; EXTERN;
0132
0133            [ASYNCHRONOUS] FUNCTION LIB$SET_SCROLL (
0134
0135                            VAR START_LINE,
0136                                END_LINE    : [VOLATILE] INTEGER
0137
0138                                ) : INTEGER; EXTERN;
0139
0140            FUNCTION LIB$PUT_SCREEN (
0141
0142                        %STDESCR TEXTX      : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0143                            VAR LINE,
0144                                COLUMN      : INTEGER
0145
0146                                ) : INTEGER; EXTERN;
0147
0148            [UNBOUND,EXTERNAL] FUNCTION LIB$PUT_OUTPUT (
```

```
0149
0150                    %STDESCR TEXTX       : PACKED ARRAY [L..U:INTEGER] OF CHAR
0151
0152                        ) : INTEGER; EXTERNAL;
0153
0154            [UNBOUND,EXTERNAL] FUNCTION LIB$GET_INPUT (
0155
0156                    %DESCR   INPUT_TEXT  : VARYING [A] OF CHAR;
0157                    %DESCR   PROMPT_STR  : VARYING [B] OF CHAR;
0158                         VAR LENGTH      : INTEGER
0159
0160                        ) : INTEGER; EXTERNAL;
0161
0162            FUNCTION LIB$GET_SCREEN (
0163
0164                    %DESCR   INPUT_TEXT  : VARYING [A] OF CHAR;
0165                    %DESCR   PROMPT_STR  : VARYING [B] OF CHAR;
0166                         VAR LENGTH      : INTEGER
0167
0168                        ) : INTEGER; EXTERN;
0169
0170            FUNCTION LIB$SCREEN_INFO (
0171
0172                            FLAGS               : SCR1$TYPE;
0173                            DEV_TYPE            : INTEGER;
0174                            LINE_WIDTH          : INTEGER;
0175                            LINES_PER_PAGE      : INTEGER
0176
0177                            ) : INTEGER; EXTERN;
0178
0179            FUNCTION LIB$DOWN_SCROLL : INTEGER; EXTERN;
0180
0181            { +
0182            The following is the famed "TPARSE" table-driven parsing routine.
0183            - }
0184            FUNCTION LIB$TPARSE (
0185
0186                            VAR PARAM_BLK    : [VOLATILE] TPA$TYPE;
0187                        %IMMED STATE_TBL     : INTEGER;
0188                        %IMMED KEY_TBL       : INTEGER
0189
0190                            ) : INTEGER; EXTERN;
0191
0192            { +
0193            The following are general string handling routines.
0194            - }
0195
0196            PROCEDURE STR$UPCASE (
0197
0198                            VAR DST          : [VOLATILE] DESCRIPTOR;
0199                            VAR SRC          : [VOLATILE] DESCRIPTOR
0200
0201                            ); EXTERN;
0202
0203            PROCEDURE STR$TRIM (
0204
0205                            VAR DST          : [VOLATILE] DESCRIPTOR;
```

```
0206                %STDESCR SRC          : PACKED ARRAY [L..U:INTEGER] OF CHAR
0207
0208                    ); EXTERN;
0209
0210            [ASYNCHRONOUS] PROCEDURE STR$FREE1_DX (
0211
0212                VAR DSC               : [VOLATILE] DESCRIPTOR
0213
0214                    ); EXTERN;
0215
0216            [ASYNCHRONOUS] FUNCTION LIB$SCOPY_DXDX  (
0217
0218                    SRC               : DESCRIPTOR;
0219                    DST               : DESCRIPTOR
0220
0221                    ) : INTEGER; EXTERN;
0222
0223            FUNCTION OTS$SCOPY_DXDX (
0224
0225                %STDESCR SRC          : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0226                    DST               : DESCRIPTOR
0227
0228                    ) : INTEGER; EXTERN;
0229
0230            FUNCTION OTS$CVT_TI_L (
0231
0232                    STR               : DESCRIPTOR;
0233                VAR NUM               : INTEGER
0234
0235                    ) : INTEGER; EXTERN;
0236
0237            FUNCTION LIB$DATE_TIME (
0238
0239                %STDESCR DST          : PACKED ARRAY [L..U:INTEGER] OF CHAR
0240
0241                    ) : INTEGER; EXTERN;
0242
0243            { +
0244            This routine will set a bit field (SRC) in a location (BASE).
0245            - }
0246            [ASYNCHRONOUS] PROCEDURE LIB$INSV (
0247
0248                    SRC,
0249                    POS               : LONG;
0250                    SIZE              : BYTE;
0251                VAR BASE              : LONG
0252
0253                    ); EXTERN;
0254
0255            { +
0256            This routine will get a bit field from a location.
0257            - }
0258            [ASYNCHRONOUS] FUNCTION LIB$EXTZV (
0259
0260                    POS,
0261                    SIZE,
0262                    BASE              : LONG
```

```
                        ) : INTEGER; EXTERN;

        { +
        This routine does a schdwk/hiber to wait a specified length of time.
        - }
        [ASYNCHRONOUS] PROCEDURE LIB$WAIT (

                        SECONDS     : REAL

                        ); EXTERN;

        { +
        This routine does our help processing for us.
        - }
        FUNCTION LBR$OUTPUT_HELP (

                    %IMMED PUT_PTR              : INTEGER;
                           WIDTH                : LONG;
                    %IMMED LINE                 : LONG;
                  %STDESCR LIBRARY              : PACKED ARRAY [L..U:INTEGER] OF CHAR;
                    %IMMED FLAGS                : LONG;
                    %IMMED GET_PTR              : INTEGER

                        ) : INTEGER; EXTERNAL;

        { +
        These are used in error handling.
        - }

        [ASYNCHRONOUS] FUNCTION LIB$MATCH_COND (

                        CONDVAL,
                        COMPVAL     : INTEGER

                        ) : BOOLEAN; EXTERN;

        PROCEDURE LIB$SIGNAL (

                    %IMMED CONDITION    : INTEGER;
                    %IMMED NUMARGS      : INTEGER;
                    %IMMED ARG1         : INTEGER;
                    %IMMED ARG2         : INTEGER

                        ); EXTERN;

        PROCEDURE LIB$STOP (

                    %IMMED CONDITION    : INTEGER;
                    %IMMED NUMARGS      : INTEGER;
                    %IMMED ARG1         : INTEGER;
                    %IMMED ARG2         : INTEGER

                        ); EXTERN;

        { +
        These are used to process the DCL command string.
```

```
0320            - }
0321
0322            FUNCTION CLI$GET_VALUE (
0323
0324                        %STDESCR SELECT      : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0325                             VAR RETURNED    : DESCRIPTOR
0326
0327                                 ) : INTEGER; EXTERN;
0328
0329            FUNCTION CLI$PRESENT (
0330
0331                        %STDESCR TEXTX       : PACKED ARRAY [L..U:INTEGER] OF CHAR
0332
0333                                 ) : INTEGER; EXTERN;
0334
0335
0336            { +
0337            The following is used to plot graphs on the terminal.
0338            - }
0339            PROCEDURE EDF$GRAPH (
0340
0341                                 GRAPH_TYPE   : INTEGER;
0342                          %DESCR XY_ARRAY     : XY_PLOT_TYPE;
0343                                 CURRENT_INDEX,
0344                                 LAST_INDEX,
0345                                 Y_HIGH,
0346                                 Y_LOW,
0347                                 Y_INCR       : INTEGER;
0348                        %STDESCR Y_LABEL      : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0349                          %DESCR COLOR_ARRAY  : XY_PLOT_TYPE
0350
0351                                 ) ; EXTERN;
0352
0353            { +
0354            The following is used to parse the input FDL file for storage into the
0355            database.
0356            - }
0357            FUNCTION FDL$PARSE (
0358
0359                                 FNM          : DESCRIPTOR;
0360                           VAR FAB            : FAB_PTR;
0361                           VAR RAB            : FAB_PTR;
0362                          %REF FLG            : FDL2$TYPE
0363
0364                                 ) : INTEGER; EXTERN;
0365
0366            { +
0367            The following is used to deallocate the memory used by the RMS blocks
0368            which are allocated by FDL$PARSE.
0369            - }
0370            FUNCTION FDL$RELEASE (
0371
0372                                 FAB          : FAB_PTR;
0373                                 RAB          : FAB_PTR
0374
0375                                 ) : INTEGER; EXTERN;
0376
```

```
0377            { +
0378            The following is used to first parse an input FDL file and then create the
0379            RMS file specified by it.
0380            - }
0381            FUNCTION FDL$CREATE (
0382
0383                                FDLNAM      : DESCRIPTOR;
0384                                OUTNAM      : DESCRIPTOR;
0385                                DFTNAM      : DESCRIPTOR;
0386                              VAR RESNAM    : [VOLATILE] DESCRIPTOR;
0387                              VAR FIDBLK    : [VOLATILE] FID;
0388                             %REF FLAGS     : FDL2$TYPE
0389
0390                                ) : INTEGER; EXTERN;
0391
0392            { +
0393            The following routines are used to interface to TTDRV.
0394            - }
0395            FUNCTION  EDF$TERM_SETUP : INTEGER; EXTERN;
0396
0397            PROCEDURE EDF$RESET_TERM; EXTERN;
0398
0399            PROCEDURE EDF$CTRLCAST; EXTERN;
0400
0401            { +
0402            The following routines are in other modules of EDF.
0403            They were FORWARD declared before seperate compilation (Pascal V2) was used.
0404            - }
0405            [EXTERNAL] FUNCTION QUERY (OFFSET : INTEGER) : BOOLEAN; EXTERN;
0406
0407            [EXTERNAL] PROCEDURE POINT_AT_DEFINITION; EXTERN;
0408
0409            [ASYNCHRONOUS,EXTERNAL] PROCEDURE EDF$RESET_SCROLL; EXTERN;
0410
0411            [EXTERNAL] PROCEDURE INPUT_ANALYSIS_FILE; EXTERN;
0412
0413            [EXTERNAL] PROCEDURE SHOW_ALL_PRIMARIES; EXTERN;
0414
0415            [EXTERNAL] PROCEDURE SHOW_PRIMARY_SECTION (TEST : LINE_OBJECT); EXTERN;
0416
0417            [EXTERNAL] PROCEDURE SHOW_CUR_PRI_SEC (PLUS_VALUE : BOOLEAN); EXTERN;
0418
0419            [EXTERNAL] FUNCTION NATURAL_DEPTH : INTEGER; EXTERN;
0420
0421            END.
0422            {       End of file SRC$:EDFEXTERN.PAS      }
```

```
                        .TITLE  EDFEXTERN
                        .IDENT  \V04-000\

00000                   .END
```

COMMAND QUALIFIERS

```
  PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS$:EDFEXTERN/OBJ=OBJ$:EDFEXTERN MSRC$:EDFEXTERN

  /CHECK=(NOBOUNDS,NOCASE_SELECTORS,NOOVERFLOW,NOPOINTERS,NOSUBRANGE)
  /DEBUG=(NOSYMBOLS,NOTRACEBACK)
  /ENVIRONMENT= $255$DUA28:[EDF.OBJ]EDFEXTERN.PEN;1
  /LIST= $255$DUA28:[EDF.LIS]EDFEXTERN.LIS;1
  /OBJECT= $255$DUA28:[EDF.OBJ]EDFEXTERN.OBJ;1
  /NOCROSS_REFERENCE /ERROR_LIMIT=30 /NOG_FLOATING /MACHINE_CODE /NOOLD_VERSION /OPTIMIZE /NOSTANDARD /WARNINGS
```

COMPILER INTERNAL TIMING

Phase	Faults	CPU Time	Elapsed Time
Initialization	71	00:00.4	00:01.9
Source Analysis	572	00:10.1	02:31.9
Source Listing	43	00:00.7	00:02.4
Tree Construction	0	00:00.0	00:00.0
Flow Analysis	0	00:00.0	00:00.0
Profit Analysis	0	00:00.0	00:00.0
Context Analysis	0	00:00.0	00:00.0
Name Packing	0	00:00.0	00:00.0
Code Selection	0	00:00.0	00:00.0
Final	92	00:00.7	00:01.9
TOTAL	780	00:11.9	02:38.2

COMPILATION STATISTICS

```
  CPU Time:       00:11.9         (2130 Lines/Minute)
  Elapsed Time:   02:38.2
  Page Faults:    780
  Compilation Complete
```

EDFCHF
LIS

EDFFUNCS
LIS

EDFEXTERN
LIS

EDFDESIGN
LIS

EDFCONST
LIS