

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	FFFFFFFFFFFFFFFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEEEEEEEEEEEEEEE	DDD	FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDD	FFFFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDD	FFFFFFFFFFFFFFFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEE	DDD	FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	FFF

5
Va
--
00
00
00
00
00
00
00
00
00
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	EE	XX	XX	TT	EE	RR	NN	NN
EE	DD	DD	EE	XX	XX	TT	EE	RR	NN	NN
EE	DD	DD	EE	XX	XX	TT	EE	RR	NN	NN
EEEEEEEEEE	DD	DD	EEEEEEEEEE	XX	XX	TT	EEEEEEEEEE	RRRRRRRR	NN	NN
EEEEEEEEEE	DD	DD	EEEEEEEEEE	XX	XX	TT	EEEEEEEEEE	RRRRRRRR	NN	NN
EE	DD	DD	EE	XX	XX	TT	EE	RR	NN	NN
EE	DD	DD	EE	XX	XX	TT	EE	RR	NN	NN
EE	DD	DD	EE	XX	XX	TT	EE	RR	NN	NN
EE	DD	DD	EE	XX	XX	TT	EE	RR	NN	NN
EEEEEEEEEE	DDDDDDDD	FF	EEEEEEEEEE	XX	XX	TT	EEEEEEEEEE	RR	NN	NN
EEEEEEEEEE	DDDDDDDD	FF	EEEEEEEEEE	XX	XX	TT	EEEEEEEEEE	RR	NN	NN

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
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      *****
```

0030 FACILITY: VAX/VMS EDF (EDIT/FDL) UTILITY

0031 ABSTRACT: This facility is used to create, modify, and optimize
0032 FDL specification files.

0033 ENVIRONMENT: NATIVE/USER MODE

0034 AUTHOR: Ken F. Henderson Jr.

0035 CREATION DATE: 27-Mar-1981

0036 MODIFIED BY:

0037	V03-008	KFH0008	Ken Henderson	8 Aug 1983
0038			Changes for seperate compilation.	
0039	V03-007	KFH0007	Ken Henderson	14 Apr 1983
0040			Changed EDF\$TERM_SETUP function	
0041			value from BYTE to INTEGER.	
0042	V03-006	KFH0006	Ken Henderson	22 Nov 1982
0043			Removed obsolete SYS\$routine	
0044			definitions. (now from starlet.pen)	
0045	V03-005	KFH0005	Ken Henderson	11 Oct 1982
0046			Added support for Access, ACL,	
0047			Connect, Journal, Sharing primaries.	
0048			Changed FDL\$\$PARSE to FDL\$PARSE.	
0049				
0050				
0051				
0052				
0053				
0054				
0055				
0056				
0057				

EDFEXTERN
V04-000

L 13
16-Sep-1984 00:45:41
5-Sep-1984 13:36:55

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFEXTERN.PAS;1 (1) 2

Source Listing

Modified FDL\$CREATE.

V03-004 KFH0004 Ken Henderson 22 Sept 1982
Added definition of LIB\$WAIT;
Removed definition of FOR\$SECNDS.

V03-003 KFH0003 Ken Henderson 11 August 1982
Removed reference to CLISEND_PARSE.

V03-002 KFH0002 Ken Henderson 2 April 1982
Added ISECNUM_PTR to call to
FDL\$GLOBAL.

V03-001 KFH0001 Ken Henderson 23-Mar-1982
Modified LIB\$OUTPUT_HELP to fix
FT2 QAR 381

-- }

EDFEXTERN
V04-000

Source Listing

M 13
16-Sep-1984 00:45:41
5-Sep-1984 13:36:55

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFEXTERN.PAS;1 (2) Page 3

```
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$PUT_LINE (
0105     %DESCR      TEXTX      : [VOLATILE] VARYING [A] OF CHAR;
0106     VAR LINE_ADV : [VOLATILE] INTEGER;
0107     FLAGS        : [VOLATILE] INTEGER
0108 ) : INTEGER; EXTERNAL;
0109
0110 FUNCTION LIB$ERASE_LINE (
0111     VAR LINE,
0112     COLUMN      : INTEGER
0113 ) : INTEGER; EXTERN;
0114
0115 FUNCTION LIB$ERASE_PAGE (
0116     VAR LINE,
0117     COLUMN      : [VOLATILE] INTEGER
0118 ) : INTEGER; EXTERN;
0119
0120 FUNCTION LIB$SET_CURSOR (
0121     VAR LINE,
0122     COLUMN      : [VOLATILE] INTEGER
0123 ) : INTEGER; EXTERN;
0124
0125 FUNCTION LIB$SET_SCROLL (
0126     VAR START_LINE,
0127     END_LINE      : [VOLATILE] INTEGER
0128 ) : INTEGER; EXTERN;
0129
0130 [ASYNCHRONOUS] FUNCTION LIB$SET_SCROLL (
0131     VAR START_LINE,
0132     END_LINE      : [VOLATILE] INTEGER
0133 ) : INTEGER; EXTERN;
0134
0135 FUNCTION LIB$PUT_SCREEN (
0136     %STDSCR TEXTX      : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0137     VAR LINE,
0138     COLUMN      : INTEGER
0139 ) : INTEGER; EXTERN;
0140
0141 [UNBOUND,EXTERNAL] FUNCTION LIB$PUT_OUTPUT (
```



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



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



```
0263
0264           ) : INTEGER; EXTERN;
0265
0266 { +
0267 This routine does a schdwk/hiber to wait a specified length of time.
0268 - }
0269 [ASYNCHRONOUS] PROCEDURE LIB$WAIT (
0270
0271         SECONDS      : REAL
0272
0273 ); EXTERN;
0274
0275 { +
0276 This routine does our help processing for us.
0277 - }
0278 FUNCTION LBR$OUTPUT_HELP (
0279
0280         %IMMED PUT_PTR      : INTEGER;
0281         %IMMED WIDTH        : LONG;
0282         %IMMED LINE         : LONG;
0283         %STDSCR LIBRARY     : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0284         %IMMED FLAGS        : LONG;
0285         %IMMED GET_PTR      : INTEGER
0286
0287 ) : INTEGER; EXTERNAL;
0288
0289 { +
0290 These are used in error handling.
0291 - }
0292
0293 [ASYNCHRONOUS] FUNCTION LIB$MATCH_COND (
0294
0295         CONDVAL,
0296         COMPVAL      : INTEGER
0297
0298 ) : BOOLEAN; EXTERN;
0299
0300 PROCEDURE LIB$SIGNAL (
0301
0302         %IMMED CONDITION      : INTEGER;
0303         %IMMED NUMARGS        : INTEGER;
0304         %IMMED ARG1           : INTEGER;
0305         %IMMED ARG2           : INTEGER
0306
0307 ); EXTERN;
0308
0309 PROCEDURE LIB$STOP (
0310
0311         %IMMED CONDITION      : INTEGER;
0312         %IMMED NUMARGS        : INTEGER;
0313         %IMMED ARG1           : INTEGER;
0314         %IMMED ARG2           : INTEGER
0315
0316 ); EXTERN;
0317
0318 { +
0319 These are used to process the DCL command string.
```

```
0320 - )
0321
0322 FUNCTION CLISGET_VALUE (
0323
0324     %STDSCR SELECT      : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0325     VAR RETURNED       : DESCRIPTOR
0326
0327     ) : INTEGER; EXTERN;
0328
0329 FUNCTION CLISPRESENT (
0330
0331     %STDSCR TEXTX       : PACKED ARRAY [L..U:INTEGER] OF CHAR
0332
0333     ) : INTEGER; EXTERN;
0334
0335 { +
0336 The following is used to plot graphs on the terminal.
0337 - )
0338
0339 PROCEDURE EDF$GRAPH (
0340
0341     GRAPH_TYPE : INTEGER;
0342     %DESCR XY_ARRAY : XY_PLOT_TYPE;
0343     CURRENT_INDEX,
0344     LAST_INDEX,
0345     Y_HIGH,
0346     Y_LOW,
0347     Y_INCR      : INTEGER;
0348     %STDSCR Y_LABEL : PACKED ARRAY [L..U:INTEGER] OF CHAR;
0349     %DESCR COLOR_ARRAY : XY_PLOT_TYPE
0350
0351     ) : EXTERN;
0352
0353 { +
0354 The following is used to parse the input FDL file for storage into the
0355 database.
0356 - )
0357 FUNCTION FDL$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$CTRLCAST; EXTERN;
0400
0401 { +
0402 The following routines are in other modules of EDF.
0403 They were FORWARD declared before sepearte compilation (Pascal V2) was used.
0404 - }
0405 [EXTERNAL] FUNCTION QUERY (OFFSET : INTEGER) : BOOLEAN; EXTERN;
0406
0407 [EXTERNAL] PROCEDURE POINT_AT_DEFINITION; EXTERN;
0408
0409 [ASYNCHRONOUS,EXTERNAL] PROCEDURE EDF$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      }
```

Generated Code

G 14
16-Sep-1984 00:45:41
5-Sep-1984 13:36:55

VAX-11 Pascal V2.4-277 Page 10
DISK\$VMSMASTER:[EDF.SRC]EDFEXTERN.PAS;1 (3)

.TITLE EDFEXTERN
.IDENT \V04-000\

00000

.END

EDFEXTERN
V04-000

Pascal Compilation Statistics

H 14
16-Sep-1984 00:45:41
5-Sep-1984 13:36:55

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFEXTERN.PAS;1 (3) Page 11

COMMAND QUALIFIERS

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

/CHECK=(NOBOUNDS, NOCASE_SELECTORS, NOOVERFLOW, NOPPOINTERS, 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

0126

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY