

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

EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFFFFFFFFFFFFF
EEE	DDD	DDD	FFF
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
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFF
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD		FFF

```
EEEEEEEEEE DDDDDDD FFFFFFFF MM MM AAAAAA IIIIII NN NN
EEEEEEEEEE DDDDDDD FFFFFFFF MM MM AAAAAA IIIIII NN NN
EE DD DD FF MMMM MMMM AA AA II NN NN
EE DD DD FF MMMM MMMM AA AA II NN NN
EE DD DD FF MM MM AA AA II NNNN NN
EEEEEEEE DD DD FFFFFFFF MM MM AA AA II NNNN NN
EEEEEEEE DD DD FFFFFFFF MM MM AA AA II NN NN
EE DD DD FF MM MM AAAAAAAAAA II NN NNNN
EE DD DD FF MM MM AAAAAAAAAA II NN NNNN
EE DD DD FF MM MM AA AA II NN NN
EE DD DD FF MM MM AA AA II NN NN
EEEEEEEEEE DDDDDDD FFFFFFFF MM MM AA AA IIIIII NN NN
EEEEEEEEEE DDDDDDD FFFFFFFF MM MM AA AA IIIIII NN NN
```

```
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
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 IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS
```

Vertical barcode-like pattern on the right edge of the page.

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057

```

[ IDENT ('V04-000'),
( ++
*****
**
** COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
** DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
** ALL RIGHTS RESERVED.
**
** THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
** ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
** INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
** COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
** OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
** TRANSFERRED.
**
** THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
** AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
** CORPORATION.
**
** DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
** SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
**
*****

```

```

FACILITY:      VAX/VMS EDF (EDIT/FDL) UTILITY
ABSTRACT:      This facility is used to create, modify, and optimize
                FDL specification files.
ENVIRONMENT:   NATIVE/USER MODE
AUTHOR:        Ken F. Henderson Jr.
CREATION DATE: 27-Mar-1981
MODIFIED BY:
V03-011 KF0011      Ken Henderson      8 Aug 1983
                Changes for seperate compilation.
V03-010 KF0010      Ken Henderson      26 Apr 1983
                Added ADD_KEY, DELETE_KEY scripts.
                Transferred some initializations from
                INIT_EDITOR to EDFVAR. Changed 'redesign'
                to 'touchup'.
V03-009 KF0009      Ken Henderson      14 Apr 1983
                Added SET_FUNCTION, RESPONSES,
                GRANULARITY, PROMPTING, JOURNAL_ENABLED.
V03-008 KF0008      Ken Henderson      20 Jan 1983
                Removed references to DASH.

```



```
0171 EDF$AB COMMENT
0172 LIB$PUT_OUTPUT_PTR
0173 LIB$GET_INPUT_PTR
0174 DEF_CURRENT
0175 DEF_HEAD
0176 DEF_TAIL
0177 DEF_PRED
0178 DEF_SUCC
0179 INPUT_FILENAME_DESC
0180 OUTPUT_FILENAME_DESC
0181 ANALYSIS_FILENAME_DESC
0182 IDATA[EDF$K_FIRST_SCRIPT]
0183 FULL_PROMPT
0184
0185 ROUTINES CALLED:
0186
0187 EDF$TERM_SETUP
0188 ESTABLISH
0189 LIB$SIGNAL
0190 CLISGET_VALUE
0191 CLISPRESENT
0192
0193 ROUTINE VALUE:
0194
0195 none
0196
0197 SIGNALS:
0198
0199 EDF$_SMALLPAGE - if term screen size too small
0200
0201 SIDE EFFECTS:
0202
0203 EDF is initialized.
0204
0205 -- }
```

```
0207 PROCEDURE INIT_EDITOR;
0208
0209 VAR
0210     TEMP_DESCRIPTOR      : DESCRIPTOR;
0211     I                    : INTEGER;
0212
0213 BEGIN
0214
0215     ( +
0216     See if the user wants batch mode or interactive.
0217     - )
0218     IF NOT (ODD (CLISPRESENT ('INTERACTIVE')))) THEN
0219
0220     ( +
0221     At this point, the user has specified /NOINTERACTIVE and
0222     wants a quick, automatic tuneup for his file.
0223     - )
0224     BEGIN
0225
0226         TAKE_DEFAULTS                := TRUE;
0227         AUTO_TUNE                     := TRUE;
0228         JOURNAL_ENABLED               := FALSE;
0229         QTAB[EDFSK_RETURN].DEFAULT_OK := TRUE;
0230         IDATA[EDFSK_RESPONSES]       := EDFSK_AUTO;
0231         IDATA[EDFSK_FIRST_SCRIPT]    := EDFSK_OPTIMIZE_FDL;
0232         QTAB[EDFSK_CURRENT_FUNCTION].DEFAULT := EDFSK_QUIT;
0233         QTAB[EDFSK_DESIGN_CYCLE].DEFAULT := EDFSK_FINIS;
0234         VIDEO_TERMINAL                := FALSE;
0235         DEC_CRT                       := FALSE;
0236         ANSI_CRT                      := FALSE;
0237         REGIS                         := FALSE;
0238
0239     END ( IF TRUE NOT (ODD (CLISPRESENT ('INTERACTIVE')))) )
0240
0241     ELSE
0242
0243     BEGIN
0244
0245     ( +
0246     Do initialization on the terminal. Get its speed, setup exit handler,
0247     Also check to make sure that indeed the
0248     input is a terminal (and STOPS if not) and if the terminal isn't a scope,
0249     then it sets the page length to 16
0250     (as required by hardcopy surface plots).
0251
0252     *****
0253     The call to EDF$TERM_SETUP Must come BEFORE ANY calls to the
0254     SCREEN PACKAGE!!!
0255     *****
0256
0257     - )
0258     TERMINAL_SPEED := EDF$TERM_SETUP;
0259
0260     ( +
0261     EDF$TERM_SETUP returns a status of EDF$DEVCLASS if SYSS$INPUT is
0262     not a terminal. In that case, see if the magic logical name
0263     EDF$$PLAYBACK_INPUT is defined. If so, then set the terminal speed
```



```
0264 to be 2400 baud and continue, if the logical is not defined,
0265 exit with the DEVCLASS status.
0266 NOTE THAT THE LOGICAL NAME 'EDF$$PLAYBACK_INPUT' IS NOT SUPPORTED
0267 FOR CUSTOMERS AND IS USED INTERNALLY ONLY FOR REGRESSION TESTING!!!
0268 - )
0269 IF LIB$MATCH_COND (TERMINAL_SPEED,EDF$_DEVCLASS) THEN
0270
0271 BEGIN
0272
0273     TEMP_STATUS := $TRNLOG ('EDF$$PLAYBACK_INPUT',TEMP_STRING255);
0274
0275     IF LIB$MATCH_COND (TEMP_STATUS,SS$_NOTRAN) THEN
0276
0277         LIB$STOP (EDF$_DEVCLASS,0,0,0)
0278
0279     ELSE IF LIB$MATCH_COND (TEMP_STATUS,SS$_NORMAL) THEN
0280
0281         TERMINAL_SPEED := TT$_BAUD_2400;
0282
0283 END;
0284
0285 { +
0286 Set up ^C ast routine.
0287 - )
0288 EDF$CTRLCAST;
0289
0290 { +
0291 First find out what terminal characteristics we have.
0292 - )
0293 LIB$SCREEN_INFO (
0294     SCREEN_FLAGS,
0295     DEV_TYPE,
0296     LINE_WIDTH,
0297     LINES_PER_PAGE
0298 );
0299
0300 { +
0301 Instead of using the actual terminal type,
0302 we'll just test on decprt or not.
0303 - )
0304 VIDEO_TERMINAL      := SCREEN_FLAGS.SCR$_SCREEN;
0305 DEC_CRT             := SCREEN_FLAGS.SCR$_DECCRT;
0306 ANSICRT             := SCREEN_FLAGS.SCR$_ANSICRT;
0307 REGIS               := SCREEN_FLAGS.SCR$_REGIS;
0308
0309 { +
0310 See if we have enough room on the terminal.
0311 - )
0312 IF (
0313 (LINE_WIDTH < MINIMUM_TERM_WIDTH)
0314 OR
0315 (VIDEO_TERMINAL AND (LINES_PER_PAGE < MINIMUM_VIDEO_PAGE))
0316 ) THEN
0317
0318     { +
0319 Not enough room!
0320 - )
```

```
0321     LIBSSIGNAL (EDF$_SMALLPAGE,2,LINE_WIDTH,LINES_PER_PAGE);
0322
0323     OPEN      (OUTPUT,SYS$OUTPUT_NAME,NEW,RECORD_LENGTH := 2148);
0324     REWRITE (OUTPUT);
0325
0326     ( +
0327     Open the journal file, if it's requested.
0328     NOTE THAT THE LOGICAL NAME 'EDF$$JOURNAL_INPUT' IS NOT SUPPORTED FOR
0329     CUSTOMERS, AND IS USED INTERNALLY ONLY FOR REGRESSION TESTING.
0330     - )
0331     TEMP_STATUS      := $STRNLOG ('EDF$$JOURNAL_INPUT',,JOURNAL_FILENAME);
0332
0333     JOURNAL_ENABLED := (
0334                       (LIB$MATCH_COND (TEMP_STATUS,SS$_NORMAL))
0335                       AND
0336                       (NOT LIB$MATCH_COND (TEMP_STATUS,SS$_NOTRAN))
0337                       );
0338
0339     IF JOURNAL_ENABLED THEN
0340     BEGIN
0341
0342         OPEN      (
0343             FILE_VARIABLE := JOURNAL_FILE,
0344             FILE_NAME     := JOURNAL_FILENAME,
0345             HISTORY       := NEW,
0346             RECORD_LENGTH := 255,
0347             RECORD_TYPE  := VARIABLE
0348         );
0349         REWRITE   (JOURNAL_FILE);
0350
0351     END;
0352
0353     END;      ( IF FALSE (NOT ODD (CLI$PRESENT ('INTERACTIVE')))) )
0354
0355     ( +
0356     If we don't have an AN^I terminal (VT100-series and up), then zero out the
0357     video attribute arrays.
0358     - )
0359     IF NOT DEC_CRT THEN
0360     BEGIN
0361
0362         FOR I := 1 TO 4 DO
0363         BEGIN
0364             ANSI_RESET[I]      := NULL_CHAR;
0365             ANSI_BOLD[I]       := NULL_CHAR;
0366             ANSI_UNDERSCORE[I] := NULL_CHAR;
0367             ANSI_BLINK[I]      := NULL_CHAR;
0368             ANSI_REVERSE[I]    := NULL_CHAR;
0369
0370         END;
0371
0372     END;
0373
0374     END;      ( IF NOT DEC_CRT )
0375
0376
0377
```

```
0378 { +
0379 If we have more than 80 chars per line, we may have to shift everything
0380 over to the right. (if we also have a video terminal)
0381 - }
0382 IF (LINE_WIDTH > EDF$C_SHIFTPOINT) AND VIDEO_TERMINAL THEN
0383
0384 BEGIN
0385
0386     SHIFT[2]           := TAB;
0387     SHIFT[3]           := TAB;
0388     SHIFT[4]           := TAB;
0389     CRLF_SHIFT[4]      := TAB;
0390     CRLF_SHIFT[5]      := TAB;
0391     CRLF_SHIFT[6]      := TAB;
0392
0393 END;
0394
0395 { +
0396 The 'under-graph' text comes out in graphics mode for Regis devices.
0397 - }
0398 IF REGIS THEN
0399
0400 BEGIN
0401
0402     LOW_SHIFT[1]       := NULL_CHAR;
0403     LOW_SHIFT[2]       := NULL_CHAR;
0404     LOW_SHIFT[3]       := NULL_CHAR;
0405
0406 END      { IF TRUE REGIS }
0407
0408 ELSE
0409
0410 BEGIN
0411
0412     LOW_SHIFT[1]       := SHIFT[2];
0413     LOW_SHIFT[2]       := SHIFT[3];
0414     LOW_SHIFT[3]       := SHIFT[4];
0415
0416 END;      { IF FALSE REGIS }
0417
0418 { +
0419 OK, so let the user know that we're here.
0420 - }
0421 CLEAR (SCREEN);
0422
0423 { +
0424 Initialize the TPARSE block.
0425 - }
0426 WITH PARAM_BLOCK DO
0427
0428 BEGIN
0429
0430     TPASL_COUNT      := TPASK_COUNTO;
0431     TPASV_ABBREV     := TRUE;
0432
0433 END;      { DO }
```

```
0435 { +
0436 Stuff the pointer variable FDL_BLOCK with the address of FDL$AL_BLOCK
0437 PLUS the Contents of FDL$AL_BLOCK. The offset is introduced by the
0438 transfer vector in the shareable image FDL$HR.EXE. (home of FDL$AL_BLOCK)
0439 - }
0440 FDL_BLOCK::INTEGER := IADDRESS (FDL$AL_BLOCK) + FDL$AL_BLOCK;
0441
0442 { +
0443 Now stuff the address of our EDF$LINE_PARSED routine into the callback
0444 address cell in the FDL$AL_BLOCK array.
0445 - }
0446 FDL_BLOCK^[FDL$AL_PCALL] := IADDRESS (EDF$LINE_PARSED);
0447
0448 { +
0449 Setup some defaults.
0450 - }
0451 IDATA[EDF$K_RESPONSES] := EDF$K_MAN;
0452 BDATA[EDF$K_BLOCK_SPAN] := TRUE;
0453 IDATA[EDF$K_BUCKET_WEIGHT] := EDF$K_FLATTER_FILES;
0454
0455 { +
0456 This initializes the the QTAB table with the addresses of the TParse tables.
0457 - }
0458 QTAB[EDF$K_CARR_CTRL].KEY_TABLE := IADDRESS (EDF$AB_CARR_TABLE_KEY);
0459 QTAB[EDF$K_CARR_CTRL].STATE_TABLE := IADDRESS (EDF$AB_CARR_TABLE_STA);
0460 QTAB[EDF$K_RECORD_FORMAT].KEY_TABLE := IADDRESS (EDF$AB_FORMAT_TABLE_KEY);
0461 QTAB[EDF$K_RECORD_FORMAT].STATE_TABLE := IADDRESS (EDF$AB_FORMAT_TABLE_STA);
0462 QTAB[EDF$K_KEY_TYPE].KEY_TABLE := IADDRESS (EDF$AB_TYPE_TABLE_KEY);
0463 QTAB[EDF$K_KEY_TYPE].STATE_TABLE := IADDRESS (EDF$AB_TYPE_TABLE_STA);
0464 QTAB[EDF$K_LOAD_METHOD].KEY_TABLE := IADDRESS (EDF$AB_LOAD_METHOD_TABLE_KEY);
0465 QTAB[EDF$K_LOAD_METHOD].STATE_TABLE := IADDRESS (EDF$AB_LOAD_METHOD_TABLE_STA);
0466 QTAB[EDF$K_BUCKET_WEIGHT].KEY_TABLE := IADDRESS (EDF$AB_WEIGHT_TABLE_KEY);
0467 QTAB[EDF$K_BUCKET_WEIGHT].STATE_TABLE := IADDRESS (EDF$AB_WEIGHT_TABLE_STA);
0468 QTAB[EDF$K_SURFACE_OPTION].KEY_TABLE := IADDRESS (EDF$AB_SURFACE_OPTION_TABLE_KEY);
0469 QTAB[EDF$K_SURFACE_OPTION].STATE_TABLE := IADDRESS (EDF$AB_SURFACE_OPTION_TABLE_STA);
0470 QTAB[EDF$K_CURRENT_FUNCTION].KEY_TABLE := IADDRESS (EDF$AB_CURRENT_FUNC_TABLE_KEY);
0471 QTAB[EDF$K_CURRENT_FUNCTION].STATE_TABLE := IADDRESS (EDF$AB_CURRENT_FUNC_TABLE_STA);
0472 QTAB[EDF$K_DESIGN_CYCLE].KEY_TABLE := IADDRESS (EDF$AB_DESIGN_CYCLE_TABLE_KEY);
0473 QTAB[EDF$K_DESIGN_CYCLE].STATE_TABLE := IADDRESS (EDF$AB_DESIGN_CYCLE_TABLE_STA);
0474 QTAB[EDF$K_SCRIPT_OPTION].KEY_TABLE := IADDRESS (EDF$AB_SCRIPT_OPTION_TABLE_KEY);
0475 QTAB[EDF$K_SCRIPT_OPTION].STATE_TABLE := IADDRESS (EDF$AB_SCRIPT_OPTION_TABLE_STA);
0476 QTAB[EDF$K_KEY_COMP_WANTED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0477 QTAB[EDF$K_KEY_COMP_WANTED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0478 QTAB[EDF$K_REC_COMP_WANTED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0479 QTAB[EDF$K_REC_COMP_WANTED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0480 QTAB[EDF$K_IDX_COMP_WANTED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0481 QTAB[EDF$K_IDX_COMP_WANTED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0482 QTAB[EDF$K_CONFIRM].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0483 QTAB[EDF$K_CONFIRM].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0484 QTAB[EDF$K_BLOCK_SPAN].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0485 QTAB[EDF$K_BLOCK_SPAN].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0486 QTAB[EDF$K_ASCENDING_ADDED].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0487 QTAB[EDF$K_ASCENDING_ADDED].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0488 QTAB[EDF$K_ASCENDING_LOAD].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0489 QTAB[EDF$K_ASCENDING_LOAD].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
0490 QTAB[EDF$K_RETURN].KEY_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_KEY);
0491 QTAB[EDF$K_RETURN].STATE_TABLE := IADDRESS (EDF$AB_YES_NO_TABLE_STA);
```



```
0513 { +
0514 Create an Ident line_object and put it into the list.
0515 - }
0516 NEW_IDENT_LINE;
0517
0518 { +
0519 See what we have.
0520 - }
0521 ANALYSIS_SPECIFIED := ODD (CLIS$PRESENT ('ANALYSIS'));
0522
0523 IF ANALYSIS_SPECIFIED THEN
0524
0525 BEGIN
0526
0527 { +
0528 Save the analysis filename.
0529 - }
0530 ANALYSIS_FILENAME_DESC := NULL_STRING;
0531 CLIS$GET_VALUE ('ANALYSIS',ANALYSIS_FILENAME_DESC);
0532
0533 END;      ( IF ANALYSIS_SPECIFIED )
0534
0535 { +
0536 Save the input filename.
0537 - }
0538 INPUT_FILENAME_DESC := NULL_STRING;
0539 CLIS$GET_VALUE ('P1',INPUT_FILENAME_DESC);
0540
0541 { +
0542 Find out which output filename we're using, the /OUTPUT, or the
0543 command parameter.
0544 - }
0545 OUTPUT_FILENAME_DESC      := NULL_STRING;
0546
0547 IF ODD (CLIS$PRESENT ('OUTPUT')) THEN
0548
0549 BEGIN
0550
0551 { +
0552 The /OUTPUT switch overrides, so use it if present.
0553 - }
0554 CLIS$GET_VALUE ('OUTPUT',OUTPUT_FILENAME_DESC);
0555
0556 END
0557
0558 ELSE
0559
0560 { +
0561 The user just wants another version of the input file.
0562 - }
0563 LIB$SCOPY_DXX (INPUT_FILENAME_DESC,OUTPUT_FILENAME_DESC);
0564
0565 { +
0566 The following qualifiers make sense only if we're in normal
0567 interactive mode.
0568 - }
0569 IF ODD (CLIS$PRESENT ('INTERACTIVE')) THEN
```

```
0570
0571 BEGIN
0572
0573 { +
0574 Set up the script to the one specified in the DCL command. (if any)
0575 - )
0576 IF ODD (CLISPRESNT('SCRIPT')) THEN
0577
0578 BEGIN
0579
0580 TEMP_DESCRIPTOR := NULL_STRING;
0581 CLISGET_VALUE ('SCRIPT',TEMP_DESCRIPTOR);
0582
0583 { +
0584 Case on the 1st letter.
0585 - )
0586 CASE TEMP_DESCRIPTOR.DSCSA_POINTER^[1] OF
0587
0588 'A' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_ADD_KEY_FDL;
0589
0590 'D' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_DELETE_KEY_FDL;
0591
0592 'I' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_IDX_DESIGN_FDL;
0593
0594 'S' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_SEQ_DESIGN_FDL;
0595
0596 'O' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_OPTIMIZE_FDL;
0597
0598 'R' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_REL_DESIGN_FDL;
0599
0600 'T' : IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_REDESIGN_FDL;
0601
0602 OTHERWISE
0603
0604 { +
0605 If the user blows it, give him nothing.
0606 - )
0607 IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_ZERO_SCRIPT;
0608
0609 END; { CASE }
0610
0611 STR$FREE1_DX (TEMP_DESCRIPTOR);
0612
0613 END { IF TRUE SCRIPT PRESENT }
0614
0615 ELSE
0616
0617 IDATA[EDFSK_FIRST_SCRIPT] := EDFSK_ZERO_SCRIPT;
0618
0619 { +
0620 Find out how many keys the user wants.
0621 - )
0622 IF ODD (CLISPRESNT('NUMBER_KEYS')) THEN
0623
0624 BEGIN
0625
0626 TEMP_DESCRIPTOR := NULL_STRING;
```

```
0627 CLISGET_VALUE ('SCRIPT',TEMP_DESCRIPTOR);
0628 ISTATUS := OTSSCVT Till (TEMP_DESCRIPTOR,
0629 IDATA[EDFSK_NUMBER_KEYS]);
0630 QTAB[EDFSK_NUMBER_KEYS].DEFAULT := IDATA[EDFSK_NUMBER_KEYS];
0631 NUMBER_KEYS_SET := TRUE;
0632
0633 STR$FREE1_DX (TEMP_DESCRIPTOR);
0634
0635 END; ( IF TRUE SCRIPT PRESENT )
0636
0637 { +
0638 If the user specified a prompt level, set EDF's level to that,
0639 otherwise set it according to the type of terminal (hardcopy gets Brief).
0640 - )
0641 IF ODD (CLISPRESENT ('PROMPTING')) THEN
0642
0643 BEGIN
0644
0645 { +
0646 Get the prompting level specified by the user.
0647 - )
0648 TEMP_DESCRIPTOR := NULL STRING;
0649 CLISGET_VALUE ('PROMPTING',TEMP_DESCRIPTOR);
0650
0651 { +
0652 The 1st character of the string is unique.
0653 - )
0654 CASE TEMP_DESCRIPTOR.DSC$A_POINTER^[1] OF
0655
0656 { +
0657 Brief prompting
0658 - )
0659 'B' :
0660
0661 FULL_PROMPT := FALSE;
0662
0663 { +
0664 Full prompting
0665 - )
0666 'F' :
0667
0668 FULL_PROMPT := TRUE;
0669
0670 OTHERWISE
0671
0672 { +
0673 Automatic prompting.
0674 Default to Brief prompting for non-scope (or slow) terminals.
0675 - )
0676 IF ( VIDEO_TERMINAL
0677 AND
0678 ( TERMINAL_SPEED >= TTSC_BAUD_2400 ) ) THEN
0679
0680 FULL_PROMPT := TRUE
0681
0682 ELSE
0683
```



```
0798
0799           ELSE
0800
0801             IDATA[EDFSK_GRANULARITY] := EDFSK_TWO;
0802
0803           OTHERWISE
0804             ( NULL-STATEMENT ) ;
0805
0806           END; ( CASE )
0807
0808         END ( IF ODD (CLISPRESNT ('GRANULARITY')) )
0809
0810       ELSE
0811
0812         IDATA[EDFSK_GRANULARITY]      := EDFSK_THREE;
0813
0814       IF ODD (CLISPRESNT ('EMPHASIS')) THEN
0815
0816       BEGIN
0817
0818         CLISGET_VALUE ('EMPHASIS',TEMP_DESCRIPTOR);
0819
0820         CASE TEMP_DESCRIPTOR.DSCSA_POINTER^[1] OF
0821
0822           'F' :      IDATA[EDFSK_BUCKET_WEIGHT] := EDFSK_FLATTER_FILES;
0823           'S' :      IDATA[EDFSK_BUCKET_WEIGHT] := EDFSK_SMALLER_BUFFERS;
0824
0825           OTHERWISE
0826             ( NULL-STATEMENT ) ;
0827
0828           END; ( CASE )
0829
0830           QTAB[EDFSK_BUCKET_WEIGHT].DEFAULT := IDATA[EDFSK_BUCKET_WEIGHT];
0831
0832         END ( IF ODD (CLISPRESNT ('EMPHASIS')) )
0833
0834       ELSE
0835
0836         IDATA[EDFSK_BUCKET_WEIGHT]      := EDFSK_FLATTER_FILES;
0837
0838         DEFAULT_FILENAME_DESC           := NULL_STRING;
0839         STRSTRIM (DEFAULT_FILENAME_DESC, '.FDL');
0840
0841         NL_DEV_DESC := NULL_STRING;
0842         STRSTRIM (NL_DEV_DESC, 'NL:');
0843
0844         ( +
0845         Set the main loop variable to true so we can execute the main cycle.
0846         - )
0847         EDITING           := TRUE;
0848
0849       END; ( INIT_EDITOR )
0850
0851
```

```
0853      ( ++
0854
0855      INPUT_FDL_FILE -- Uses FDL$PARSE to read the user's input FDL file.
0856
0857      This routine parses the input file using FDL$PARSE.
0858
0859      CALLING SEQUENCE:
0860
0861      INPUT_FDL_FILE:
0862
0863      INPUT PARAMETERS:
0864
0865      none
0866
0867      IMPLICIT INPUTS:
0868
0869      none
0870
0871      OUTPUT PARAMETERS:
0872
0873      none
0874
0875      IMPLICIT OUTPUTS:
0876
0877      The Definition Linked List
0878
0879      ROUTINES CALLED:
0880
0881      FDL$PARSE
0882
0883      ROUTINE VALUE:
0884
0885      none
0886
0887      SIGNALS:
0888
0889
0890      SIDE EFFECTS:
0891
0892      none
0893
0894      -- )
```

```
0896 PROCEDURE INPUT_FDL_FILE;
0897
0898 BEGIN
0899
0900     ( +
0901     Set up the condition handler for the disk.
0902     - )
0903     ESTABLISH (RMS_INPUT_COND_HANDLER);
0904
0905     ( +
0906     Now tell the user what we're doing.
0907     - )
0908     IF NOT AUTO_TUNE THEN
0909         WRITELN (SHIFT,TAB,TAB,'Parsing Definition File');
0910
0911     ( +
0912     Make sure edf$line_parsed gets the non-analysis stuff.
0913     - )
0914     ANALYSIS_ONLY      := FALSE;
0915
0916     ( +
0917     Turn on the $CALLBACK flags bit to make FDL$PARSE call us.
0918     Also, turn on the SIGNAL bit to make FDL$PARSE signal errors to us.
0919     - )
0920     FLAGS.FDL$V_$CALLBACK      := TRUE;
0921     FLAGS.FDL$V_$SIGNAL        := TRUE;
0922
0923     ( +
0924     Parse the input file to get the old definition.
0925     If the input file doesn't exist, this will signal up to the main loop.
0926     - )
0927     ISTATUS      := FDL$PARSE (
0928         INPUT_FILENAME_DESC,
0929         FAB_DUMMY,
0930         RAB_DUMMY,
0931         FLAGS
0932     );
0933
0934     IF (
0935         (ODD (ISTATUS))
0936         AND
0937         (NOT AUTO_TUNE)
0938     ) THEN
0939         WRITELN (SHIFT,TAB,TAB,'Definition Parse Complete');
0940
0941     END; ( INPUT_FDL_FILE )
0942
0943
```

```
0945      ( ++
0946
0947      INPUT_ANALYSIS_FILE -- Read in the analysis file if specified.
0948
0949      This routine parses the user's analysis file if he has specified one.
0950
0951      CALLING SEQUENCE:
0952
0953      INPUT_ANALYSIS_FILE:
0954
0955      INPUT PARAMETERS:
0956
0957      none
0958
0959      IMPLICIT INPUTS:
0960
0961      none
0962
0963      OUTPUT PARAMETERS:
0964
0965      none
0966
0967      IMPLICIT OUTPUTS:
0968
0969      The Analysis Linked List
0970
0971      ROUTINES CALLED:
0972
0973      FDL$PARSE
0974
0975      ROUTINE VALUE:
0976
0977      none
0978
0979      SIGNALS:
0980
0981
0982      SIDE EFFECTS:
0983
0984      none
0985
0986      -- )
```

```
0988 [GLOBAL] PROCEDURE INPUT_ANALYSIS_FILE;  
0989  
0990 BEGIN  
0991     { +  
0992     Only do this if the user wants to.  
0993     - }  
0994     IF EDITING AND ANALYSIS_SPECIFIED THEN  
0995     BEGIN  
0996  
0997     BEGIN  
0998         { +  
0999         Set up the condition handler for the disk.  
1000         - }  
1001         ESTABLISH (RMS_INPUT_COND_HANDLER);  
1002  
1003         { +  
1004         Now tell the user what we're doing.  
1005         - }  
1006         IF NOT AUTO_TUNE THEN  
1007             WRITELN (SHIFT, 'Parsing Analysis File');  
1008  
1009         { +  
1010         Make sure edf$line_parsed gets only the analysis stuff.  
1011         - }  
1012         ANALYSIS_ONLY := TRUE;  
1013         POINT_AT_ANALYSIS;  
1014  
1015         { +  
1016         Create an Ident line_object and put it into the list.  
1017         - }  
1018         NEW_IDENT_LINE;  
1019  
1020         { +  
1021         Turn on the $CALLBACK flags bit to make FDL$PARSE call us.  
1022         Also, turn on the SIGNAL bit to make FDL$PARSE signal errors to us.  
1023         - }  
1024         FLAGS.FDL$V_$CALLBACK := TRUE;  
1025         FLAGS.FDL$V_$SIGNAL := TRUE;  
1026  
1027         { +  
1028         Parse the analysis file to get the analysis sections.  
1029         If it doesn't exist, this will signal up to the main loop.  
1030         - }  
1031         ISTATUS := FDL$PARSE (  
1032             ANALYSIS_FILENAME_DESC,  
1033             FAB_DUMMY,  
1034             RAB_DUMMY,  
1035             FLAGS  
1036         );  
1037  
1038         ANALYSIS_ONLY := FALSE;  
1039         POINT_AT_DEFINITION;  
1040  
1041         IF (  
1042             ODD (ISTATUS))  
1043         )
```

EDF
V04-000

Source Listing

M 11
16-Sep-1984 01:22:54
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277 Page 22
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS,1 (10)

```
1045 AND
1046 (NOT AUTO_TUNE)
1047 ) THEN
1048
1049     WRITELN (SHIFT,'Analysis Parse Complete',CRLF);
1050
1051 END:      { IF EDITING AND ANALYSIS_SPECIFIED }
1052
1053 END:      { INPUT_ANALYSIS_FILE }
```


EDF
V04-000

Source Listing

N 11
16-Sep-1984 01:22:54
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (11) Page 23

```
1055      ( ++
1056
1057      SETUP_CONTINUE -- Get ready to ask the user if he wants to continue after
1058      an input parse error.
1059
1060      CALLING SEQUENCE:
1061
1062      SETUP_CONTINUE:
1063
1064      INPUT PARAMETERS:
1065
1066      none
1067
1068      IMPLICIT INPUTS:
1069
1070      none
1071
1072      OUTPUT PARAMETERS:
1073
1074      none
1075
1076      IMPLICIT OUTPUTS:
1077
1078      none
1079
1080      ROUTINES CALLED:
1081
1082      CLEAR
1083
1084      ROUTINE VALUE:
1085
1086      none
1087
1088      SIGNALS:
1089
1090      none
1091
1092      SIDE EFFECTS:
1093
1094      none
1095      -- }
1096
```

EDF
V04-000

Source Listing

B 12
16-Sep-1984 01:22:54
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (12) Page 24

EDI
V04

1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109

```
PROCEDURE SETUP_CONTINUE;  
BEGIN  
    ( +  
    Set up the control/Z handler in case the user types ^Z.  
    - )  
    ESTABLISH (CTRLZ_COND_HANDLER);  
    CLEAR (PAUSE);  
END;    ( SETUP_CONTINUE )
```

```
1111 ( **
1112
1113 DISPATCH_FUNCTION -- Branch off to the selected function.
1114
1115 This routine is just a big CASE statement to execute the operation the
1116 user wants.
1117
1118 CALLING SEQUENCE:
1119
1120 DISPATCH_FUNCTION
1121
1122 INPUT PARAMETERS:
1123
1124 none
1125
1126 IMPLICIT INPUTS:
1127
1128 IDATA[EDFSK_FIRST_SCRIPT]
1129 IDATA[EDFSK_CURRENT_FUNCTION]
1130
1131 OUTPUT PARAMETERS:
1132
1133 none
1134
1135 IMPLICIT OUTPUTS:
1136
1137 EDITING
1138 IDATA[EDFSK_SCRIPT_OPTION]
1139
1140 ROUTINES CALLED:
1141
1142 ADD_FDL_LINE
1143 DELETE_FDL_LINE
1144 CREATE_NEW_FDL
1145 HELP_PROC
1146 MODIFY_FDL_LINE
1147 INVOKE_SCRIPT
1148 VIEW_DEF
1149
1150 ROUTINE VALUE:
1151
1152 none
1153
1154 SIGNALS:
1155
1156
1157 SIDE EFFECTS:
1158
1159 none
1160
1161 -- )
```

```
1163 PROCEDURE DISPATCH_FUNCTION;
1164
1165 BEGIN
1166     { +
1167     Set up the control/Z handler and reinitialize some flags.
1168     - }
1169     ESTABLISH (CTRLZ_COND_HANDLER);
1170
1171     IF NOT AUTO_TUNE THEN
1172         CLOSE (FDL_DEST,ERROR := CONTINUE);
1173
1174     POINT AT DEFINITION;
1175     DEST_IS_TERMINAL := TRUE;
1176     OPTIMIZING := FALSE;
1177     VISIBLE_QUESTION := FALSE;
1178     TEMP_FUCL_PROMPT := FALSE;
1179     TAKE_DEFAULTS := AUTO_TUNE;
1180
1181     { +
1182     Ask the user only if he hadn't requested one from DCL.
1183     - }
1184     IF IDATA[EDFSK_FIRST_SCRIPT] = EDFSK_ZERO_SCRIPT THEN
1185         BEGIN
1186             { +
1187             Get the user's top-level function and dispatch on it.
1188             - }
1189             QUERY (EDFSK_CURRENT_FUNCTION);
1190
1191             CASE IDATA[EDFSK_CURRENT_FUNCTION] OF
1192                 EDFSK_ADD :      ADD_FDL_LINE; { Add a new line_object to the list. }
1193                 EDFSK_DELETE :  DELETE_FDL_LINE; { Remove a line_object from the list. }
1194                 EDFSK_HELP :    HELP_PROC; { Prompt for help and process it. }
1195                 EDFSK_INVOKE :  INVOKE_SCRIPT; { Ask a bunch of related questions. }
1196                 EDFSK_MODIFY :  MODIFY_FDL_LINE; { Edit an extant line_object. }
1197                 EDFSK_QUIT :    EDITING := FALSE; { Wipe out! All bets are off! }
1198                 EDFSK_SET :     SET_PROC; { Set the editor characteristics. }
1199                 EDFSK_VIEW :    VIEW_DEF; { Show the user the definition. }
1200
1201                 EDFSK_EXIT :
1202                     BEGIN
1203                         { +
1204                         Stop the editing loop and output the new FDL file.
1205                         - }
1206                         EDITING := FALSE;
1207                         CREATE_NEW_FDL;
1208
1209                     END;
1210
1211             END;
1212
1213         END;
1214
1215     END;
1216
```

```
1218      OTHERWISE
1219
1220          { NULL-STATEMENT } ;
1221
1222      END;      { CASE }
1223
1224  END          { IF TRUE IDATA[EDF$K_FIRST_SCRIPT] = EDF$K_ZERO_SCRIPT }
1225
1226  ELSE
1227
1228  BEGIN
1229
1230      { +
1231      The user wants to do a script right off, do it.
1232      - }
1233      IDATA[EDF$K_SCRIPT_OPTION]      := IDATA[EDF$K_FIRST_SCRIPT];
1234
1235      ISAM_ORG      := (IDATA[EDF$K_SCRIPT_OPTION] IN [ EDF$K_ADD_KEY_FDL,
1236      EDF$K_DELETE_KEY_FDL, EDF$K_IDX_DESIGN_FDL,
1237      EDF$K_REDESIGN_FDL, EDF$K_OPTIMIZE_FDL ]);
1238
1239      INVOKE_SCRIPT;
1240
1241  END;          { IF FALSE IDATA[EDF$K_FIRST_SCRIPT] = EDF$K_ZERO_SCRIPT }
1242
1243  END;      { DISPATCH_FUNCTION }
```

```
1245 ( **
1246 +-----+
1247 | *** THIS IS THE TOP LEVEL CODE IN THE UTILITY. *** |
1248 +-----+
1249 -- )
1250
1251 BEGIN
1252
1253 ( +
1254 Set up the editor, setup the exit and condition handlers, a control/C
1255 AST routine, and get all the DCL switch options.
1256 Set EDITING to TRUE.
1257 - )
1258 INIT_EDITOR;
1259
1260 ( +
1261 Read in the FDL file, and possibly an analysis file.
1262 1st clear the error flag.
1263 - )
1264 RMS_INPUT_ERROR := FALSE;
1265
1266 IF NOT NO_INPUT THEN
1267     INPUT_FDL_FILE;
1268
1269 ( +
1270 If we had an error, pause to let the user read the messages,
1271 otherwise, continue on.
1272 - )
1273 IF EDITING AND RMS_INPUT_ERROR THEN
1274     SETUP_CONTINUE
1275
1276 ELSE IF NOT NO_INPUT THEN
1277     LIB$WAIT (3.0);
1278
1279
1280
1281
```

EDF
V04-000

Source Listing

G 12
16-Sep-1984 01:22:54
5-Sep-1984 13:37:22

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFMAIN.PAS;1 (17) Page 29

```
1283      { +
1284      This is the Main Loop.
1285      - }
1286      WHILE EDITING DO
1287
1288      BEGIN
1289
1290          DISPATCH_FUNCTION;
1291
1292          IF MAIN_CTRLZ THEN
1293
1294              BEGIN
1295
1296                  { +
1297                  Stop the editing loop and output the new FDL file.
1298                  - }
1299                  EDITING      := FALSE;
1300                  CREATE_NEW_FDL;
1301
1302              END;
1303
1304          END;      ( WHILE EDITING )
1305
1306      { +
1307      Close the journal file if we had one.
1308      - }
1309      IF JOURNAL_ENABLED THEN
1310
1311          CLOSE (JOURNAL_FILE);
1312
1313      END.      ( EDF UTILITY. )
```

														.TITLE	EDF				
														.IDENT	\V04-000\				
														00000	.PSECT	\$CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2			
SF	4B	00	45	56	49	54	43	41	52	45	54	4E	49	00000	C.AAA:	.ASCII	\INTERACTIVE\<0>		
		43	41	42	59	41	4C	50	24	24	46	44	45	0000C	C.AAB:	.ASCII	\EDF\$\$PLAYBACK_INPUT\<0>		
						00		00	54	55	50	4E	49	0001A					
49	5F	4C	41	4E	52	55	4F	4A	24	24	46	44	45	00020	C.AAC:	.ASCII	\EDF\$\$JOURNAL_INPUT\<0><0>		
								00	00	54	55	50	4E	0002E					
						53	49	53	59	4C	41	4E	41	00034	C.AAD:	.ASCII	\ANALYSIS\		
						53	49	53	59	4C	41	4E	41	0003C	C.AAE:	.ASCII	\ANALYSIS\		
								00	00	31	50			00044	C.AAF:	.ASCII	\P1\<0><0>		
						00	00	54	55	50	54	55	4F	00048	C.AAG:	.ASCII	\OUTPUT\<0><0>		
		00	45	56	49	54	43	41	52	45	54	4E	49	00050	C.AAH:	.ASCII	\OUTPUT\<0><0>		
						00	00	54	50	49	52	43	53	00058	C.AAI:	.ASCII	\INTERACTIVE\<0>		
						00	00	54	50	49	52	43	53	00064	C.AAJ:	.ASCII	\SCRIPT\<0><0>		
		00	53	59	45	4B	5F	52	45	42	4D	55	4E	0006C	C.AAK:	.ASCII	\SCRIPT\<0><0>		
						00	00	54	50	49	52	43	53	00074	C.AAL:	.ASCII	\NUMBER KEYS\<0>		
		00	00	00	47	4E	49	54	50	4D	4F	52	50	00080	C.AAM:	.ASCII	\SCRIPT\<0><0>		
		00	00	00	47	4E	49	54	50	4D	4F	52	50	00088	C.AAN:	.ASCII	\PROMPTING\<0><0><0>		
						00	59	41	4C	50	53	49	44	00094	C.AAO:	.ASCII	\PROMPTING\<0><0><0>		
						00	59	41	4C	50	53	49	44	000A0	C.AAP:	.ASCII	\DISPLAY\<0>		
						00	59	41	4C	50	53	49	44	000A8	C.AAQ:	.ASCII	\DISPLAY\<0>		
		00	00	00	53	45	53	4E	4F	50	53	45	52	000B0	C.AAR:	.ASCII	\RESPONSES\<0><0><0>		
		00	00	00	53	45	53	4E	4F	50	53	45	52	000BC	C.AAS:	.ASCII	\RESPONSES\<0><0><0>		
						00	00	45	54	41	45	52	43	000C8	C.AAT:	.ASCII	\CREATE\<0><0>		
		00	59	54	49	52	41	4C	55	4E	41	52	47	000D0	C.AAU:	.ASCII	\GRANULARITY\<0>		
		00	59	54	49	52	41	4C	55	4E	41	52	47	000DC	C.AAV:	.ASCII	\GRANULARITY\<0>		
						53	49	53	41	48	50	4D	45	000E8	C.AAW:	.ASCII	\EMPHASIS\		
						53	49	53	41	48	50	4D	45	000F0	C.AAX:	.ASCII	\EMPHASIS\		
								4C	44	46	2E			000F8	C.AAY:	.ASCII	\.FDL\		
								00	3A	4C	4E			000FC	C.AAZ:	.ASCII	\NL:\<0>		
69	6E	69	66	65	44	20	67	6E	69	73	72	61	50	00100	C.ABA:	.ASCII	\Parsing Definition File\<0>		
				00	65	6C	69	6E	69	6E	6F	69	74	0010E					
72	61	50	20	6E	6F	69	74	69	6E	69	66	65	44	00118	C.ABB:	.ASCII	\Definition Parse Complete\<0><0><0>		
00	00	00	65	74	65	6C	70	6D	6F	43	20	65	73	00126					
73	79	6C	61	6E	41	20	67	6E	69	73	72	61	50	00134	C.ABC:	.ASCII	\Parsing Analysis File\<0><0><0>		
				00	00	00	65	6C	69	46	20	73	69	00142					
65	73	72	61	50	20	73	69	73	79	6C	61	6E	41	0014C	C.ABD:	.ASCII	\Analysis Parse Complete\<0>		
				00	65	74	65	6C	70	6D	6F	43	20	0015A					
		00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000067	00000000	00000000	00000000	00000000	00164	C.ABE:	.LONG	^X67,0,0,0,0,0,0,0		
						00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00178					
						00V	AF		00	FB	00002			0000	EDF:	.WORD	^M<>	: 0001	
								00000000G	EF	94	00006					CALLS	#0,INIT EDITOR	: 1259	
									00	E0	0000C					CLRB	RMS INPUT ERROR	: 1265	
						00V00000000G	EF		00	FB	00014					BBS	#0,NO INPUT,2\$: 1267	
						0000V	CF		00	E1	00019					CALLS	#0,INPUT FDL FILE	: 1269	
						00V00000000G	EF		00	E1	00021				2\$:	BBC	#0,EDITING,5\$: 1275	
						00V00000000G	EF		00	FB	00029					BBC	#0,RMS INPUT ERROR,5\$		
						0000V	CF		00V	11	0002E					CALLS	#0,SETUP_CONTINUE	: 1277	
									00V	E0	00030					BRB	12\$		
						00V00000000G	EF		00	DF	00038				5\$:	BBS	#0,NO INPUT,12\$: 1279	
						00000000G	EF	00004140	8F	DF	00038					PUSHAF	#^f3.0	: 1281	
									01	FB	0003E					CALLS	#1,LIB\$WAIT		
									00V	11	00045					BRB	12\$: 1286	

Generated Code							
0000V	CF	00	FB	00047	9\$:	CALLS	#0,DISPATCH_FUNCTION : 1290
00V00000000G	EF	00	E1	0004C		BBC	#0,MAIN_CTREZ,12\$: 1292
		00000000G	EF	94	00054	CLRB	EDITING : 1299
00000000G	EF	00	FB	0005A		CALLS	#0,CREATE_NEW_FDL : 1300
DE 00000000G	EF	00	E0	00061	12\$:	BBS	#0,EDITING,9\$
00V00000000G	EF	00	E1	00069		BBC	#0,JOURNAL_ENABLED,15\$: 1309
		00000000G	EF	9F	00071	PUSHAB	JOURNAL_FILE : 1311
00000000G	EF	01	FB	00077		CALLS	#1,PAS\$CLOSE2
	50	01	D0	0007E	15\$:	MOVL	#1,R0 : 1313
			04	00081		RET	

: Routine Size: 130 bytes, Routine Base: \$CODE + 00184

			003C	00000	INIT_EDITOR:		: 0207
			C2	00002	.WORD	^M<R2,R3,R4,R5>	
			D0	00005	SUBL2	#28,SP	
EC	AD	FFFFFDE4	8F	00000	MOVL	#17694731,-8(FP)	: 0218
		FC	0B	0000D	MOVCL	#11,C.AAA,-20(FP)	
			AD	9E	00016	MOVAB	-20(FP),-4(FP)
			AD	9F	0001B	PUSHAB	-8(FP)
			01	FB	0001E	CALLS	#1,CLIS\$PRESENT
			50	E8	00025	BLBS	R0,2\$
			01	90	00028	MOVB	#1,TAKE_DEFAULTS : 0226
			01	90	0002F	MOVB	#1,AUTO_TUNE : 0227
			EF	94	00036	CLRB	JOURNAL_ENABLED : 0228
			01	90	0003C	MOVB	#1,QTAB+504 : 0229
			EF	D4	00043	CLRL	IDATA+260 : 0230
			05	D0	00049	MOVL	#5,IDATA+8 : 0231
			06	D0	00050	MOVL	#6,QTAB+755 : 0232
			01	D0	00057	MOVL	#1,QTAB+780 : 0233
			EF	94	0005E	CLRB	VIDEO_TERMINAL : 0234
			EF	94	00064	CLRB	DEC CRT : 0235
			EF	94	0006A	CLRB	ANST CRT : 0236
			EF	94	00070	CLRB	REGIS : 0237
			0000V	31	00076	BRW	17\$
			00	FB	00079	CALLS	#0,EDF\$TERM_SETUP : 0258
			50	D0	00080	MOVL	R0,TERMINAL_SPEED
			8F	DF	00087	PUSHAL	#11763724 : 0269
			EF	9F	0008D	PUSHAB	TERMINAL_SPEED
			02	FB	00093	CALLS	#2,LIB\$MATCH_COND
			50	E8	0009A	BLBS	R0,..+3
			0000V	31	0009D	BRW	9\$
			00	DD	000A0	PUSHL	#0 : 0273
			00	DD	000A2	PUSHL	#0
			00	DD	000A4	PUSHL	#0
			8F	D0	000A6	MOVL	#17694975,-8(FP)
			EF	9E	000AE	MOVAB	TEMP_STRING255,-4(FP)
			AD	9F	000B6	PUSHAB	-8(FP)
			00	DD	000B9	PUSHL	#0
			8F	D0	000BB	MOVL	#17694739,-16(FP)
			EF	9E	000C3	MOVAB	C.AAB,-12(FP)
			AD	9F	000CB	PUSHAB	-16(FP)
			06	FB	000CE	CALLS	#6,SYS\$TRNLOG
			50	D0	000D5	MOVL	R0,TEMP_STATUS
			8F	DF	000DC	PUSHAL	#1577 : 0275
			EF	9F	000E2	PUSHAB	TEMP_STATUS
			02	FB	000E8	CALLS	#2,LIB\$MATCH_COND

Generated Code

J 12
16-Sep-1984 01:22:54
5-Sep-1984 13:37:22

		00V	50	E9	000EF	BLBC	R0,5\$	
			00	DD	000F2	PUSHL	#0	: 0277
			00	DD	000F4	PUSHL	#0	
			00	DD	000F6	PUSHL	#0	
		00B3800C	8F	DD	000F8	PUSHL	#11763724	
		00000000G	EF	04	FB	000FE	CALLS	#4,LIB\$STOP
			00V	11	00105	BRB	9\$	
		00000001	8F	DF	00107	5\$: PUSHAL	#1	: 0279
		00000000G	EF	9F	0010D	PUSHAB	TEMP STATUS	
		00000000G	EF	02	FB	00113	CALLS	#2,LIB\$MATCH_COND
			00V	50	E9	0011A	BLBC	R0,9\$
		00000000G	EF	0B	D0	0011D	MOVL	#11,TERMINAL_SPEED
		00000000G	EF	00	FB	00124	9\$: CALLS	#0,EDF\$CTRLCAST
			00000000G	EF	9F	0012B	PUSHAB	LINES PER PAGE
			00000000G	EF	9F	00131	PUSHAB	LINE WIDTH
			00000000G	EF	9F	00137	PUSHAB	DEV TYPE
			00000000G	EF	9F	0013D	PUSHAB	SCREEN_FLAGS
		00000000G	EF	04	FB	00143	CALLS	#4,LIB\$SCREEN_INFO
		00000000G	EF	90	0014A	MOVAB	SCREEN_FLAGS,VIDEO_TERMINAL	: 0304
50	00000000G	EF	01	06	EF	00155	EXTZV	#6,#1,SCREEN_FLAGS,R0
	00000000G	EF	50	90	0015E	MOVAB	R0,DEC CRT	: 0305
50	00000000G	EF	01	01	EF	00165	EXTZV	#1,#1,SCREEN_FLAGS,R0
	00000000G	EF	50	90	0016E	MOVAB	R0,ANSI CRT	: 0306
50	00000000G	EF	01	02	EF	00175	EXTZV	#2,#1,SCREEN_FLAGS,R0
	00000000G	EF	50	90	0017E	MOVAB	R0,REGIS	: 0307
	00000000G	EF	00V	D1	00185	CMPL	LINE_WIDTH,MINIMUM_TERM_WIDTH	: 0312
	00V00000000G	EF	00	E1	00192	BLSS	13\$	
	00000000G	EF	00V	D1	0019A	BBF	#0,VIDEO_TERMINAL,14\$	
		00000000G	EF	18	001A5	CMPL	LINES_PER_PAGE,MINIMUM_VIDEO_PAGE	
		00000000G	EF	DD	001A7	BGEQ	14\$: 0321
		00000000G	EF	DD	001AD	PUSHL	LINES PER PAGE	
			02	DD	001B3	PUSHL	LINE_WIDTH	
		00B38014	8F	DD	001B5	PUSHL	#2	
		00000864	EF	04	FB	001BB	PUSHL	#11763732
			8F	DD	001C2	CALLS	#4,LIB\$SIGNAL	: 0323
			07	DD	001C8	PUSHL	#2,48	
			04	DD	001CA	PUSHL	#7	
		00000000G	EF	9F	001CC	PUSHL	#4	
			0B	DD	001D2	PUSHAB	SYSS\$OUTPUT_NAME	
			01	DD	001D4	PUSHL	#11	
		00000000G	EF	9F	001D6	PUSHL	#1	
		00000000G	EF	07	FB	001DC	PUSHAB	PASS\$V OUTPUT
			EF	9F	001E3	CALLS	#7,PASS\$OPEN2	: 0324
		00000000G	EF	01	FB	001E9	PUSHAB	PASS\$V OUTPUT
			00	DD	001F0	CALLS	#1,PASS\$REWRITE2	: 0331
			00	DD	001F2	PUSHL	#0	
			00	DD	001F4	PUSHL	#0	
		F8 AD 010E00FF	8F	D0	001F6	PUSHL	#0	
		FC AD 00000000G	EF	9E	001FE	MOVL	#17694975,-8(FP)	
			AD	9F	00206	MOVAB	JOURNAL_FILENAME,-4(FP)	
			00	DD	00209	PUSHAB	-8(FP)	
		F0 AD 010E0012	8F	D0	0020B	PUSHL	#0	
		F4 AD FFFFFFFBFF	EF	9E	00213	MOVL	#17694738,-16(FP)	
			AD	9F	0021B	MOVAB	C.AAC,-12(FP)	
		00000000G	EF	06	FB	0021E	PUSHAB	-16(FP)
		00000000G	EF	50	D0	00225	CALLS	#6,SYSS\$TRNLOG
						MOVL	R0,TEMP_STATUS	

Generated Code

			00000629	8F	DF	0022C	PUSHAL	#1577		: 0333	
			00000000G	EF	9F	00232	PUSHAB	TEMP_STATUS			
		00000000G		5C	FB	00238	CALLS	#2,LIBSMATCH_COND			
				50	90	0023F	MOVB	R0,R12			
			00000001	8F	DF	00242	PUSHAL	#1			
			00000000G	EF	9F	00248	PUSHAB	TEMP_STATUS			
		00000000G		5C	FB	0024E	CALLS	#2,LIBSMATCH_COND			
00000000G	EF			50	8B	00255	BICB3	R12,R0,JOURNAL_ENABLED			
		00V00000000G		EF	00	E1	0025D	BBC	#0,JOURNAL_ENABLED,16\$: 0339	
				0C	DD	00265	PUSHL	#12		: 0343	
			000000FF	8F	DD	00267	PUSHL	#255			
				07	DD	0026D	PUSHL	#7			
				04	DD	0026F	PUSHL	#4			
			00000000G	EF	9F	00271	PUSHAB	JOURNAL_FILENAME			
			000000FF	8F	DD	00277	PUSHL	#255			
				01	DD	0027D	PUSHL	#1			
		00000000G		EF	9F	0027F	PUSHAB	JOURNAL_FILE			
				08	FB	00285	CALLS	#8,PASS\$OPEN2			
		00000000G		EF	9F	0028C	PUSHAB	JOURNAL_FILE		: 0350	
				01	FB	00292	CALLS	#1,PASS\$REWRITE2			
						00299		16\$:			
		00V00000000G		EF	00	E0	00299	BBS	#0,DEC_CRT,20\$: 0360	
				50	01	50	002A1	MOVL	#1,R0	: 0364	
				5C	50	D0	002A4	MOVL	R0,I	: 0364	
			FFFFFFFFGEF4C	00000000G	EF	90	002A7	MOVB	NULL_CHAR,ANSI_RESET-1[I]	: 0368	
			FFFFFFFFGEF4C	00000000G	EF	90	002B3	MOVB	NULL_CHAR,ANSI_BOLD-1[I]	: 0369	
			FFFFFFFFGEF4C	00000000G	EF	90	002BF	MOVB	NULL_CHAR,ANSI_UNDERSCORE-1[I]	: 0370	
			FFFFFFFFGEF4C	00000000G	EF	90	002CB	MOVB	NULL_CHAR,ANSI_BLINK-1[I]	: 0371	
			FFFFFFFFGEF4C	00000000G	EF	90	002D7	MOVB	NULL_CHAR,ANSI_REVERSE-1[I]	: 0372	
				BD	04	r3	002E3	AOBLEQ	#4,R0,19\$		
00000000G	EF	64		8F	00	ED	002E7	20\$:	CMPZV	#0,#7,#^X64,LINE_WIDTH	: 0382
					00V	18	002F1	BGEQ	23\$		
		00V00000000G		EF	00	E1	002F3	BBC	#0,VIDEO_TERMINAL,23\$		
			00000001G	EF	90	002FB	MOVB	TAB,SHIFT+1		: 0386	
			00000002G	EF	90	00306	MOVB	TAB,SHIFT+2		: 0387	
			00000003G	EF	90	00311	MOVB	TAB,SHIFT+3		: 0388	
			00000003G	EF	90	0031C	MOVB	TAB,CRLF_SHIFT+3		: 0389	
			00000004G	EF	90	00327	MOVB	TAB,CRLF_SHIFT+4		: 0390	
			00000005G	EF	90	00332	MOVB	TAB,CRLF_SHIFT+5		: 0391	
		00V00000000G		EF	00	E1	0033D	23\$:	BBC	#0,REGIS,25\$: 0398
			00000000G	EF	90	00345	MOVB	NULL_CHAR,LOW_SHIFT		: 0402	
			00000001G	EF	90	00350	MOVB	NULL_CHAR,LOW_SHIFT+1		: 0403	
			00000002G	EF	90	0035B	MOVB	NULL_CHAR,LOW_SHIFT+2		: 0404	
					00V	11	00366	BRB	26\$		
			00000000G	EF	90	00368	MOVB	SHIFT+1,LOW_SHIFT		: 0412	
			00000001G	EF	90	00373	MOVB	SHIFT+2,LOW_SHIFT+1		: 0413	
			00000002G	EF	90	0037E	MOVB	SHIFT+3,LOW_SHIFT+2		: 0414	
					8F	DF	00389	26\$:	PUSHAL	#3	: 0421
			00000000G	EF	01	FB	0038F	CALLS	#1,CLEAR		
			00000000G	EF	08	D0	00396	MOVL	#8,PARAM_BLOCK	: 0430	
00000000G	EF	01		21	01	F0	0039D	INSV	#1,#33,#T,PARAM_BLOCK	: 0431	
				50	9E	003A6	MOVAB	FDLSAL_BLOCK,R0		: 0440	
		00000000G		EF	C1	003AD	ADDL3	FDLSAL_BLOCK,R0,FDL_BLOCK			
				50	D0	003B9	MOVL	FDL_BLOCK,R0		: 0446	
				04	AO	003C0	MOVAB	EDF\$LINE_PARSED,4(R0)			
			00000104G	EF	01	D0	003C8	MOVL	#1,IDATA\$260	: 0451	
			00000011G	EF	01	90	003CF	MOVB	#1,BDATA+17	: 0452	

E
V
C
C
C
C

00000098G	EF		01	DO	003D6	MOVL	#1, IDATA+152	:	0453
000002CDG	EF	00000000G	EF	9E	003D0	MOVAB	EDF\$AB_CARR_TABLE_KEY, QTAB+717	:	0458
000002D1G	EF	00000000G	EF	9E	003E8	MOVAB	EDF\$AB_CARR_TABLE_STA, QTAB+721	:	0459
0000053EG	EF	00000000G	EF	9E	003F3	MOVAB	EDF\$AB_FORMAT_TABLE_KEY, QTAB+1342	:	0460
00000542G	EF	00000000G	EF	9E	003FE	MOVAB	EDF\$AB_FORMAT_TABLE_STA, QTAB+1346	:	0461
0000045DG	EF	00000000G	EF	9E	00409	MOVAB	EDF\$AB_TYPE_TABLE_KEY, QTAB+1117	:	0462
00000461G	EF	00000000G	EF	9E	00414	MOVAB	EDF\$AB_TYPE_TABLE_STA, QTAB+1121	:	0463
00000476G	EF	00000000G	EF	9E	0041F	MOVAB	EDF\$AB_LOAD_METHOD_TABLE_KEY, QTAB+1142	:	0464
0000047AG	EF	00000000G	EF	9E	0042A	MOVAB	EDF\$AB_LOAD_METHOD_TABLE_STA, QTAB+1146	:	0465
000002B4G	EF	00000000G	EF	9E	00435	MOVAB	EDF\$AB_WEIGHT_TABLE_KEY, QTAB+692	:	0466
000002B8G	EF	00000000G	EF	9E	00440	MOVAB	EDF\$AB_WEIGHT_TABLE_STA, QTAB+696	:	0467
000005D4G	EF	00000000G	EF	9E	0044B	MOVAB	EDF\$AB_SURFACE_OPTION_TABLE_KEY, QTAB+1492	:	0468
000005D8G	EF	00000000G	EF	9E	00456	MOVAB	EDF\$AB_SURFACE_OPTION_TABLE_STA, QTAB+1496	:	0469
000002FFG	EF	00000000G	EF	9E	00461	MOVAB	EDF\$AB_CURRENT_FUNC_TABLE_KEY, QTAB+767	:	0470
00000303G	EF	00000000G	EF	9E	0046C	MOVAB	EDF\$AB_CURRENT_FUNC_TABLE_STA, QTAB+771	:	0471
00000318G	EF	00000000G	EF	9E	00477	MOVAB	EDF\$AB_DESIGN_CYCLE_TABLE_KEY, QTAB+792	:	0472
0000031CG	EF	00000000G	EF	9E	00482	MOVAB	EDF\$AB_DESIGN_CYCLE_TABLE_STA, QTAB+796	:	0473
00000570G	EF	00000000G	EF	9E	0048D	MOVAB	EDF\$AB_SCRIPT_OPTION_TABLE_KEY, QTAB+1392	:	0474
00000574G	EF	00000000G	EF	9E	00498	MOVAB	EDF\$AB_SCRIPT_OPTION_TABLE_STA, QTAB+1396	:	0475
000000D9G	EF	00000000G	EF	9E	004A3	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+217	:	0476
000000DDG	EF	00000000G	EF	9E	004AE	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+221	:	0477
000000F2G	EF	00000000G	EF	9E	004B9	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+242	:	0478
000000F6G	EF	00000000G	EF	9E	004C4	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+246	:	0479
0000010BG	EF	00000000G	EF	9E	004CF	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+267	:	0480
0000010FG	EF	00000000G	EF	9E	004DA	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+271	:	0481
0000016FG	EF	00000000G	EF	9E	004E5	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+367	:	0482
00000173G	EF	00000000G	EF	9E	004F0	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+371	:	0483
00000156G	EF	00000000G	EF	9E	004FB	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+342	:	0484
0000015AG	EF	00000000G	EF	9E	00506	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+346	:	0485
00000124G	EF	00000000G	EF	9E	00511	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+292	:	0486
00000128G	EF	00000000G	EF	9E	0051C	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+296	:	0487
0000013DG	EF	00000000G	EF	9E	00527	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+317	:	0488
00000141G	EF	00000000G	EF	9E	00532	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+321	:	0489
00000205G	EF	00000000G	EF	9E	0053D	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+517	:	0490
00000209G	EF	00000000G	EF	9E	00548	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+521	:	0491
000001ECG	EF	00000000G	EF	9E	00553	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+492	:	0492
000001F0G	EF	00000000G	EF	9E	0055E	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+496	:	0493
000001D3G	EF	00000000G	EF	9E	00569	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+467	:	0494
000001D7G	EF	00000000G	EF	9E	00574	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+471	:	0495
000001BAG	EF	00000000G	EF	9E	0057F	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+442	:	0496
000001BEG	EF	00000000G	EF	9E	0058A	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+446	:	0497
00000188G	EF	00000000G	EF	9E	00595	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+392	:	0498
0000018CG	EF	00000000G	EF	9E	005A0	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+396	:	0499
000001A1G	EF	00000000G	EF	9E	005AB	MOVAB	EDF\$AB_YES_NO_TABLE_KEY, QTAB+417	:	0500
000001A5G	EF	00000000G	EF	9E	005B6	MOVAB	EDF\$AB_YES_NO_TABLE_STA, QTAB+421	:	0501
000005EDG	EF	00000000G	EF	9E	005C1	MOVAB	EDF\$AB_PRIMARY_TABLE_KEY, QTAB+1517	:	0502
000005F1G	EF	00000000G	EF	9E	005CC	MOVAB	EDF\$AB_PRIMARY_TABLE_STA, QTAB+1521	:	0503
00000589G	EF	00000000G	EF	9E	005D7	MOVAB	EDF\$AB_SET_FUNCTION_TABLE_KEY, QTAB+1417	:	0504
0000058DG	EF	00000000G	EF	9E	005E2	MOVAB	EDF\$AB_SET_FUNCTION_TABLE_STA, QTAB+1421	:	0505
00000395G	EF	00000000G	EF	9E	005ED	MOVAB	EDF\$AB GRANULARITY_TABLE_KEY, QTAB+917	:	0506
00000399G	EF	00000000G	EF	9E	005F8	MOVAB	EDF\$AB GRANULARITY_TABLE_STA, QTAB+921	:	0507
00000525G	EF	00000000G	EF	9E	00603	MOVAB	EDF\$AB_PROMPTING_TABLE_KEY, QTAB+1317	:	0508
00000529G	EF	00000000G	EF	9E	0060E	MOVAB	EDF\$AB_PROMPTING_TABLE_STA, QTAB+1321	:	0509
00000557G	EF	00000000G	EF	9E	00619	MOVAB	EDF\$AB_RESPONSES_TABLE_KEY, QTAB+1367	:	0510
0000055BG	EF	00000000G	EF	9E	00624	MOVAB	EDF\$AB_RESPONSES_TABLE_STA, QTAB+1371	:	0511
00000000G	EF		00	FB	0062F	CALLS	#0, NEW_IDENT_LINE	:	0516
F8	AD	010E0008	8F	DO	00636	MOVL	#17694728, -8(TFP)	:	0521

Generated Code

FO	AD	FFFFF7E7 FC	EF AD	FO F8	AD AD	08 9E	28 0063E 00647	MOV C3 MOVAB	#8, C.AAD, -16(FP) -16(FP), -4(FP)	
		00000000G	EF 50	FE	01	FB	0064C 0064F	PUSHAB CALLS	-8(FP) #1, CLISPRESNT	
00000000G	EF	00V0000C000G 00000000G	EF EF		00 00	E1	00656 0065F	BICB3 BBC	#^XFE, RO, ANALYSIS SPECIFIED #0, ANALYSIS SPECIFIED, 29\$: 0523
			EF	010E0008	EF	7D	00667	MOVQ	NULL STRING, ANALYSIS_FILENAME_DESC	: 0530
			EF		EF	9F	00672	PUSHAB	ANALYSIS_FILENAME_DESC	: 0531
FO	AD	FFFFF7AD FC	EF AD	FO F8	AD AD	08 9E	00678 00680 00689	MOVL MOV C3 MOVAB	#17694728, -8(FP) #8, C.AAE, -16(FP) -16(FP), -4(FP)	
		00000000G 00000000G	EF EF		02	FB	0068E 00691	PUSHAB CALLS	-8(FP) #2, CLISGET VALUE	
			EF	00000000G	EF	7D	00698	MOVQ	NULL STRING, INPUT_FILENAME_DESC	: 0538
			EF	00000000G	EF	9F	006A3	PUSHAB	INPUT_FILENAME_DESC	: 0539
			EF	010E0002	8F	D0	006A9	MOVL	#17694722, -8(FP)	
			AD	FFFFF785	EF	B0	006B1	MOVW	C.AAF, -12(FP)	
			AD		AD	9E	006B9	MOVAB	-12(FP), -4(FP)	
			AD	F4	AD	9F	006BE	PUSHAB	-8(FP)	
			AD	F8	AD	02	FB	006C1	CALLS	#2, CLISGET VALUE
			EF	00000000G	EF	7D	006C8	MOVQ	NULL STRING, OUTPUT_FILENAME_DESC	: 0545
			EF	010E0006	8F	D0	006D3	MOVL	#17694726, -8(FP)	: 0547
FO	AD	FFFFF75E FC	EF AD	FO F8	AD AD	06 9E	006DB 006E4	MOV C3 MOVAB	#6, C.AAG, -16(FP) -16(FP), -4(FP)	
		00000000G	EF		AD	9F	006E9	PUSHAB	-8(FP)	
			00V		01	FB	006EC	CALLS	#1, CLISPRESNT	
			00V		50	E9	006F3	BLBC	RO, 33\$	
			EF	00000000G	EF	9F	006F6	PUSHAB	OUTPUT_FILENAME_DESC	: 0554
			EF	010E0006	8F	D0	006FC	MOVL	#17694726, -8(FP)	
FO	AD	FFFFF73D FC	EF AD	FO F8	AD AD	06 9E	00704 0070D	MOV C3 MOVAB	#6, C.AAH, -16(FP) -16(FP), -4(FP)	
		00000000G	EF		AD	9F	00712	PUSHAB	-8(FP)	
			EF		02	FB	00715	CALLS	#2, CLISGET_VALUE	
			00V		11	0071C		BRB	35\$	
			EF	00000000G	EF	9F	0071E	PUSHAB	OUTPUT_FILENAME_DESC	: 0563
			EF	00000000G	EF	9F	00724	PUSHAB	INPUT_FILENAME_DESC	
			EF	02	FB	0072A		CALLS	#2, LIB\$SCOPY DXDX	
			EF	010E000B	8F	D0	00731	MOVL	#17694731, -8(FP)	: 0569
EC	AD	FFFFF710 FC	EF AD	EC F8	AD AD	0B 9E	00739 00742	MOV C3 MOVAB	#11, C.AAI, -20(FP) -20(FP), -4(FP)	
		00000000G	EF		AD	9F	00747	PUSHAB	-8(FP)	
			03		01	FB	0074A	CALLS	#1, CLISPRESNT	
			03		50	E8	00751	BLBS	RO, +3	
			0000V		31	00754		BRW	87\$	
			EF	010E0006	8F	D0	00757	MOVL	#17694726, -8(FP)	: 0576
FO	AD	FFFFF6F6 FC	EF AD	FO F8	AD AD	06 9E	0075F 00768	MOV C3 MOVAB	#6, C.AAJ, -16(FP) -16(FP), -4(FP)	
		00000000G	EF		AD	9F	0076D	PUSHAB	-8(FP)	
			03		01	FB	00770	CALLS	#1, CLISPRESNT	
			03		50	E8	00777	BLBS	RO, +3	
			0000V		31	0077A		BRW	48\$	
			EF	00000000G	EF	7D	0077D	MOVQ	NULL_STRING_TEMP_DESCRIPTOR	: 0580
			EF		AD	9F	00785	PUSHAB	TEMP_DESCRIPTOR	: 0581
			EF	010E0006	8F	D0	00788	MOVL	#17694726, -16(FP)	
E8	AD	FFFFF6CD F4	EF AD	E8 FO	AD AD	06 9E	00790 00799	MOV C3 MOVAB	#6, C.AAK, -24(FP) -24(FP), -12(FP)	
			AD		AD	9F	0079E	PUSHAB	-16(FP)	

		00000000G	EF		02	FB 007A1	CALLS	#2,CLISGET VALUE		
			50		BD	9A 007A8	MOVZBL	@TEMP_DESCRIPTOR+4,RO		: 0586
	13	41	8F	FC	50	8F 007AC	CASEB	RO,#65,#19		
					0000V		.DISPL	39\$		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0000V		.DISPL	40\$		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0000V		.DISPL	41\$		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0000V		.DISPL	43\$		
					0028		.DISPL	40		
					0028		.DISPL	40		
					0000V		.DISPL	43\$		
					0028		.DISPL	40		
					0000V		.DISPL	44\$		
					0000V		.DISPL	42\$		
					0000V		.DISPL	45\$		
					00V	11 007D9	BRB	46\$		
		00000008G	EF		D4	007DB	39\$:	CLRL	IDATA+8	: 0588
					00V	11 007E1		BRB	47\$	
		00000008G	EF		01	D0 007E3	40\$:	MOVL	#1, IDATA+8	: 0590
					00V	11 007EA		BRB	47\$	
		00000008G	EF		02	D0 007EC	41\$:	MOVL	#2, IDATA+8	: 0592
					00V	11 007F3		BRB	47\$	
		00000008G	EF		04	D0 007F5	42\$:	MOVL	#4, IDATA+8	: 0594
					00V	11 007FC		BRB	47\$	
		00000008G	EF		05	D0 007FE	43\$:	MOVL	#5, IDATA+8	: 0596
					00V	11 00805		BRB	47\$	
		00000008G	EF		03	D0 00807	44\$:	MOVL	#3, IDATA+8	: 0598
					00V	11 0080E		BRB	47\$	
		00000008G	EF		06	D0 00810	45\$:	MOVL	#6, IDATA+8	: 0600
					00V	11 00817		BRB	47\$	
		00000008G	EF		07	D0 00819	46\$:	MOVL	#7, IDATA+8	: 0607
				F8	AD	9F 00820	47\$:	PUSHAB	TEMP_DESCRIPTOR	: 0611
		00000000G	EF		01	FB 00823		CALLS	#1, STR\$FREE1_DX	
					00V	11 0082A		BRB	49\$	
		00000008G	EF		07	D0 0082C	48\$:	MOVL	#7, IDATA+8	: 0617
		F0	AD	010E000B	8F	D0 00833	49\$:	MOVL	#17694731,-16(FP)	: 0622
E4	AD	FFFFFF62A	EF		08	28 0083B		MOVC3	#11,C.AAL,-28(FP)	
		F4	AD	E4	AD	9E 00844		MOVAB	-28(FP),-12(FP)	
				F0	AD	9F 00849		PUSHAB	-16(FP)	
		00000000G	EF		01	FB 0084C		CALLS	#1,CLISPRESENT	
			00V		50	E9 00853		BLBC	RO,52\$	
		F8	AD	00000000G	EF	7D 00856		MOVQ	NULL_STRING,TEMP_DESCRIPTOR	: 0626
				F8	AD	9F 0085E		PUSHAB	TEMP_DESCRIPTOR	: 0627
		F0	AD	010E0006	8F	D0 00861		MOVL	#17694726,-16(FP)	
E8	AD	FFFFFF608	EF		06	28 00869		MOVC3	#6,C.AAM,-24(FP)	
		F4	AD	E8	AD	9E 00872		MOVAB	-24(FP),-12(FP)	
				F0	AD	9F 00877		PUSHAB	-16(FP)	
		00000000G	EF		02	FB 0087A		CALLS	#2,CLISGET_VALUE	
				00000F0G	EF	9F 00881		PUSHAB	IDATA+240	: 0628

			F8	AD	9F	00887	PUSHAB	TEMP_DESCRIPTOR	
	00000000G	EF		02	FB	0088A	CALLS	#2,OTS\$CVT_TI_L	
	00000000G	EF		50	DO	00891	MOVL	R0,ISTATUS	
	000004CEG	EF	000000F0G	EF	DO	00898	MOVL	IDATA+240,QTAB+1230	: 0630
	00000000G	EF		01	90	008A3	MOVAB	#1,NUMBER_KEYS_SET	: 0631
			F8	AD	9F	008AA	PUSHAB	TEMP_DESCRIPTOR	: 0633
	00000000G	EF		01	FB	008AD	CALLS	#1,STR\$FREE1_DX	
E4	AD FFFFF5BD	AD	010E0009	8F	DO	008B4	52\$: MOVL	#17694729,-16(FP)	: 0641
	F4	AD		09	28	008BC	MOVAB	#9,C.AAN,-28(FP)	
			F8	AD	9E	008C5	MOVAB	-28(FP),-12(FP)	
			F0	AD	9F	008CA	PUSHAB	-16(FP)	
	00000000G	EF		01	FB	008CD	CALLS	#1,CLIS\$PRESENT	
		00V		50	E9	008D4	BLBC	R0,63\$	
	F8	AD	00000000G	EF	7D	008D7	MOVQ	NULL_STRING,TEMP_DESCRIPTOR	: 0648
			F8	AD	9F	008DF	PUSHAB	TEMP_DESCRIPTOR	: 0649
E4	AD FFFFF59B	AD	010E0009	8F	DO	008E2	MOVL	#17694729,-16(FP)	
	F4	AD		09	28	008EA	MOVAB	#9,C.AAO,-28(FP)	
			F0	AD	9E	008F3	MOVAB	-28(FP),-12(FP)	
			F0	AD	9F	008F8	PUSHAB	-16(FP)	
	00000000G	EF		02	FB	008FB	CALLS	#2,CLIS\$GET_VALUE	
		50		BD	9A	00902	MOVZBL	@TEMP_DESCRIPTOR+4,R0	: 0654
04	42	8F	FC	50	8F	00906	CASEB	R0,#66,#4	
				0000V		0090B	.DISPL	55\$	
				000A		0090D	.DISPL	10	
				000A		0090F	.DISPL	10	
				000A		00911	.DISPL	10	
				0000V		00913	.DISPL	56\$	
				00V	11	00915	BRB	57\$	
			00000000G	EF	94	00917	55\$: CLRB	FULL_PROMPT	: 0661
				00V	11	0091D	BRB	62\$	
	00000000G	EF		01	90	0091F	56\$: MOVAB	#1,FULL_PROMPT	: 0668
				00V	11	00926	BRB	62\$	
	00V00000000G	EF		00	E1	00928	57\$: BBC	#0,VIDEO_TERMINAL,60\$: 0676
		0B	00000000G	EF	D1	00930	CML	TERMINAL_SPEED,#11	
				00V	19	00937	BLSS	60\$	
	00000000G	EF		01	90	00939	MOVAB	#1,FULL_PROMPT	: 0680
				00V	11	00940	BRB	62\$	
			00000000G	EF	94	00942	60\$: CLRB	FULL_PROMPT	: 0684
			F8	AD	9F	00948	62\$: PUSHAB	TEMP_DESCRIPTOR	: 0688
	00000000G	EF		01	FB	0094B	CALLS	#1,STR\$FREE1_DX	
				00V	11	00952	BRB	68\$	
	00V00000000G	EF		00	E1	00954	63\$: BBC	#0,VIDEO_TERMINAL,66\$: 0699
		0B	00000000G	EF	D1	0095C	CML	TERMINAL_SPEED,#11	
				00V	19	00963	BLSS	66\$	
	00000000G	EF		01	90	00965	MOVAB	#1,FULL_PROMPT	: 0701
				00V	11	0096C	BRB	68\$	
			00000000G	EF	94	0096E	66\$: CLRB	FULL_PROMPT	: 0705
E8	AD FFFFF515	AD	010E0007	8F	DO	00974	68\$: MOVL	#17694727,-16(FP)	: 0709
	F4	AD		07	28	0097C	MOVAB	#7,C.AAP,-24(FP)	
			F8	AD	9E	00985	MOVAB	-24(FP),-12(FP)	
			F0	AD	9F	0098A	PUSHAB	-16(FP)	
	00000000G	EF		01	FB	0098D	CALLS	#1,CLIS\$PRESENT	
		03		50	E8	00994	BLBS	R0,+3	
				0000V	31	00997	BRW	79\$	
			F8	AD	9F	0099A	PUSHAB	TEMP_DESCRIPTOR	: 0713
E8	AD FFFFF4F4	AD	010E0007	8F	DO	0099D	MOVL	#17694727,-16(FP)	
		EF		07	28	009A5	MOVAB	#7,C.AAQ,-24(FP)	

Generated Code

	F4	AD		E8	AD	9E	009AE	MOVAB	-24(FP),-12(FP)		
				FO	AD	9F	009B3	PUSHAB	-16(FP)		
	00000000G	EF			02	FB	009B6	CALLS	#2,CLISGET VALUE		
		50		FC	BD	9A	009BD	MOVZBL	@TEMP_DESCRIPTOR+4,RO	: 0715	
11	41	8F			50	8F	009C1	CASEB	RO,#65,#17		
					0000V		009C6	.DISPL	73\$		
					0024		009C8	.DISPL	36		
					0024		009CA	.DISPL	36		
					0024		009CC	.DISPL	36		
					0024		009CE	.DISPL	36		
					0000V		009D0	.DISPL	72\$		
					0024		009D2	.DISPL	36		
					0024		009D4	.DISPL	36		
					0000V		009D6	.DISPL	74\$		
					0024		009D8	.DISPL	36		
					0000V		009DA	.DISPL	76\$		
					0000V		009DC	.DISPL	71\$		
					0024		009DE	.DISPL	36		
					0024		009E0	.DISPL	36		
					0024		009E2	.DISPL	36		
					0024		009E4	.DISPL	36		
					0024		009E6	.DISPL	36		
					0000V		009E8	.DISPL	75\$		
					00V	11	009EA	BRB	77\$		
	00000118G	EF			05	D0	009EC	71\$:	MOVL	#5,IDATA+280	: 0717
					00V	11	009F3		BRB	78\$	
			00000118G		EF	D4	009F5	72\$:	CLRL	IDATA+280	: 0718
					00V	11	009FB		BRB	78\$	
	00000118G	EF			03	D0	009FD	73\$:	MOVL	#3,IDATA+280	: 0719
					00V	11	00A04		BRB	78\$	
	00000118G	EF			02	D0	00A06	74\$:	MOVL	#2,IDATA+280	: 0720
					00V	11	00A0D		BRB	78\$	
	00000118G	EF			01	D0	00A0F	75\$:	MOVL	#1,IDATA+280	: 0721
					00V	11	00A16		BRB	78\$	
	00000118G	EF			04	D0	00A18	76\$:	MOVL	#4,IDATA+280	: 0722
					00V	11	00A1F		BRB	78\$	
							00A21	77\$:			
	000005C8G	EF	00000118G	EF	D0	00A21	78\$:	MOVL	IDATA+280,QTAB+1480	: 0730	
		AD	010E0009	8F	D0	00A2C	79\$:	MOVL	#17694729,-16(FP)	: 0734	
E4	AD	FFFFF46D	EF		09	28	00A34	MOV3	#9,C.AAR,-28(FP)		
		F4	AD	E4	AD	9E	00A3D	MOVAB	-28(FP),-12(FP)		
				FO	AD	9F	00A42	PUSHAB	-16(FP)		
	00000000G	EF			01	FB	00A45	CALLS	#1,CLISPRESENT		
		00V			50	E9	00A4C	BLBC	RO,86\$		
					F8	AD	9F	00A4F	PUSHAB	TEMP_DESCRIPTOR	: 0738
					010E0009	8F	D0	00A52	MOVL	#17694729,-16(FP)	
E4	AD	FFFFF453	EF		09	28	00A5A	MOV3	#9,C.AAS,-28(FP)		
		F4	AD	E4	AD	9E	00A63	MOVAB	-28(FP),-12(FP)		
				FO	AD	9F	00A68	PUSHAB	-16(FP)		
	00000000G	EF			02	FB	00A6B	CALLS	#2,CLISGET VALUE		
		50		FC	BD	9A	00A72	MOVZBL	@TEMP_DESCRIPTOR+4,RO	: 0740	
OC	41	8F			50	8F	00A76	CASEB	RO,#65,#12		
					J000V		00A7B	.DISPL	82\$		
					001A		00A7D	.DISPL	26		
					001A		00A7F	.DISPL	26		
					001A		00A81	.DISPL	26		
					001A		00A83	.DISPL	26		

				001A	00A85	.DISPL	26				
				001A	00A87	.DISPL	26				
				001A	00A89	.DISPL	26				
				001A	00A8B	.DISPL	26				
				001A	00A8D	.DISPL	26				
				001A	00A8F	.DISPL	26				
				001A	00A91	.DISPL	26				
				00COV	00A93	.DISPL	83\$				
				00V	11 00A95	BRB	84\$				
		00000104G		EF	D4 00A97	82\$: CLRL	IDATA+260		: 0742		
				00V	11 00A9D	BRB	86\$				
	00000104G	EF		01	D0 00A9F	83\$: MOVL	#1, IDATA+260		: 0743		
				00V	11 00AA6	BRB	86\$				
					00AAB	84\$:					
E8	AD	F0	AD	010E0006	8F	D0	00AAB	86\$: MOVL	#17694726,-16(FP)	: 0753	
		FFFFF409	EF		06	28	00AB0	MOV C3	#6,C.AAT,-24(FP)		
		F4	AD		E8	AD	9E	00AB9	MOV AB	-24(FP),-12(FP)	
					F0	AD	9F	00ABE	PUSH AB	-16(FP)	
						01	FB	00AC1	CALLS	#1,CLIS\$PRESENT	
00000000G	EF	00000000G	EF	50	FE	8F	8B	00AC8	BIC B3	#^XFE,RO,NO_INPUT	
					00000000G	EF	94	00AD1	CLRB	AUTO_TUNE	: 0755
						00V	11	00AD7	BRB	90\$	
				50		01	D0	00AD9	87\$: MOVL	#1,RO	: 0766
				5C		50	D0	00ADC	88\$: MOVL	RO,I	
		FFFFFFFFGEF4C	00000000G	EF	90	00ADF	MOV B	NULL_CHAR,SHIFT-1[I]		: 0770	
		00000001GEF4C	00000000G	EF	90	00AEB	MOV B	NULL_CHAR,CRLF_SHIFT+1[I]		: 0771	
E1				50		04	r3	00AF7	AOB LEQ	#4,RO,88\$	
				50		01	D0	00AFB	MOVL	#1,RO	: 0775
				5C		50	D0	00AFE	89\$: MOVL	RO,I	
		FFFFFFFFGEF4C	00000000G	EF	90	00B01	MOV B	NULL_CHAR,LOW_SHIFT-1[I]		: 0777	
ED				50		03	F3	00B0D	AOB LEQ	#3,RO,89\$	
		F0	AD	010E000B	8F	D0	00B11	90\$: MOVL	#17694731,-16(FP)	: 0781	
E4	AD	FFFFF3A8	EF		0B	28	00B19	MOV C3	#11,C.AAU,-28(FP)		
		F4	AD		E4	AD	9E	00B22	MOV AB	-28(FP),-12(FP)	
					F0	AD	9F	00B27	PUSH AB	-16(FP)	
		00000000G	EF		01	FB	00B2A	CALLS	#1,CLIS\$PRESENT		
				03		50	E8	00B31	BLBS	RO,+3	
					0000V	31	00B34	BRW	104\$		
					F8	AD	9F	00B37	PUSH AB	TEMP_DESCRIPTOR	: 0785
		F0	AD	010E000B	8F	D0	00B3A	MOVL	#17694731,-16(FP)		
E4	AD	FFFFF38B	EF		0B	28	00B42	MOV C3	#11,C.AAV,-28(FP)		
		F4	AD		E4	AD	9E	00B4B	MOV AB	-28(FP),-12(FP)	
					F0	AD	9F	00B50	PUSH AB	-16(FP)	
		00000000G	EF		02	FB	00B53	CALLS	#2,CLIS\$GET_VALUE		
				50	FC	BD	9A	00B5A	MOV ZBL	@TEMP_DESCRIPTOR+4,RO	: 0787
23				31		50	8F	00B5E	CASEB	RO,#49,#35	
				0000V				00B62	.DISPL	93\$	
				0000V				00B64	.DISPL	94\$	
				0000V				00B66	.DISPL	95\$	
				0000V				00B68	.DISPL	96\$	
				0048				00B6A	.DISPL	72	
				0048				00B6C	.DISPL	72	
				0048				00B6E	.DISPL	72	
				0048				00B70	.DISPL	72	
				0048				00B72	.DISPL	72	
				0048				00B74	.DISPL	72	
				0048				00B76	.DISPL	72	

				0048	00B78		.DISPL	72		
				0048	00B7A		.DISPL	72		
				0048	00B7C		.DISPL	72		
				0048	00B7E		.DISPL	72		
				0048	00B80		.DISPL	72		
				0048	00B82		.DISPL	72		
				0048	00B84		.DISPL	72		
				0048	00B86		.DISPL	72		
				0000V	00B88		.DISPL	97\$		
				0048	00B8A		.DISPL	72		
				0000V	00B8C		.DISPL	96\$		
				0048	00B8E		.DISPL	72		
				0048	00B90		.DISPL	72		
				0048	00B92		.DISPL	72		
				0048	00B94		.DISPL	72		
				0048	00B96		.DISPL	72		
				0048	00B98		.DISPL	72		
				0048	00B9A		.DISPL	72		
				0048	00B9C		.DISPL	72		
				0000V	00B9E		.DISPL	93\$		
				0048	00BA0		.DISPL	72		
				0048	00BA2		.DISPL	72		
				0048	00BA4		.DISPL	72		
				0048	00BA6		.DISPL	72		
				0000V	00BA8		.DISPL	98\$		
				0000V	31 00BAA		BRW	102\$		
		000000BCG		EF	D4 00BAD	93\$:	CLRL	DATA+188		: 0789
				00V	11 00BB3		BRB	105\$		
	000000BCG	EF		01	D0 00BB5	94\$:	MOVL	#1,DATA+188		: 0790
				00V	11 00BBC		BRB	105\$		
	000000BCG	EF		02	D0 00BBE	95\$:	MOVL	#2,DATA+188		: 0791
				00V	11 00BC5		BRB	105\$		
	000000BCG	EF		03	D0 00BC7	96\$:	MOVL	#3,DATA+188		: 0792
				00V	11 00BCE		BRB	105\$		
	000000BCG	EF		04	D0 00BD0	97\$:	MOVL	#4,DATA+188		: 0793
				00V	11 00BD7		BRB	105\$		
		50	FC	AD	D0 00BD9	98\$:	MOVL	TEMP_DESCRIPTOR+4,R0		: 0795
		48	8F	01	A0 91 00BDD		CMPB	1(R0),#72		
				00V	12 00BE2		BNEQ	100\$		
	000000BCG	EF		02	D0 00BE4		MOVL	#2,DATA+188		: 0797
				00V	11 00BEB		BRB	105\$		
	000000BCG	EF		01	D0 00BED	100\$:	MOVL	#1,DATA+188		: 0801
				00V	11 00BF4		BRB	105\$		
				00V	11 00BF6	102\$:	BRB	105\$		
	000000BCG	EF		02	D0 00BF8	104\$:	MOVL	#2,DATA+188		: 0813
		F0	AD	010E0008	8F	D0 00BFF	105\$:	MOVL	#17694728,-16(FP)	: 0815
E8	AD	FFFFFF2D2	EF		08	28 00C07		MOVCL	#8,C,AAW,-24(FP)	
		F4	AD		AD	9E 00C10		MOVAB	-24(FP),-12(FP)	
					AD	9F 00C15		PUSHAB	-16(FP)	
	00000000G	EF		01	FB 00C18		CALLS	#1,CLISPRESENT		
		00V		50	E9 00C1F		BLBC	R0,112\$		
			F8	AD	9F 00C22		PUSHAB	TEMP_DESCRIPTOR		: 0819
		F0	AD	010E0008	8F	D0 00C25		MOVL	#17694728,-16(FP)	
E8	AD	FFFFFF2B4	EF		08	28 00C2D		MOVCL	#8,C,AAX,-24(FP)	
		F4	AD		AD	9E 00C36		MOVAB	-24(FP),-12(FP)	
					AD	9F 00C3B		PUSHAB	-16(FP)	
	00000000G	EF		02	FB 00C3E		CALLS	#2,CLISGET_VALUE		

Generated Code

00000000G	EF	0000C000G	EF	9F	0003D	PUSHAB	PASS\$FV OUTPUT	
			03	FB	00043	CALLS	#3,PASS\$WRITE_CHAR	
			01	DD	0004A	PUSHL	#1	
	7E	00000000G	EF	9A	0004C	MOVZBL	TAB,-(SP)	
00000000G	EF	00000000G	EF	9F	00053	PUSHAB	PASS\$FV OUTPUT	
		FFFFF19B	03	FB	00059	CALLS	#3,PASS\$WRITE_CHAR	
			EF	9F	00060	PUSHAB	C.ABA	
			17	DD	00066	PUSHL	#23	
00000000G	EF	00000000G	EF	9F	00068	PUSHAB	PASS\$FV OUTPUT	
			03	FB	0006E	CALLS	#3,PASS\$WRITE_STRING	
00000000G	EF	00000000G	EF	9F	00075	PUSHAB	PASS\$FV OUTPUT	
			01	FB	0007B	CALLS	#1,PASS\$WRITELN2	
00000000G	EF	00000000G	EF	94	00082	CLR	ANALYSIS_ONLY	: 0915
			01	FD	00088	INSV	#1,#4,#1,FLAGS	: 0921
00000000G	EF		01	FD	00091	INSV	#1,#0,#1,FLAGS	: 0922
			EF	9F	0009A	PUSHAB	FLAGS	: 0928
		000000C0G	EF	9F	000A0	PUSHAB	RAB_DUMMY	
		00000000G	EF	9F	000A6	PUSHAB	FAB_DUMMY	
		00000000G	EF	9F	000AC	PUSHAB	INPUT_FILENAME_DESC	
00000000G	EF		04	FB	000B2	CALLS	#4,FD[PARSE	
00000000G	EF		50	DD	000B9	MOVL	R0,ISTATUS	
00V00000000G	EF	00V00000000G	EF	E9	000C0	BLBC	ISTATUS,5\$: 0935
			00	ED	000C7	BBS	#0,AUTO_TUNE,5\$	
			EF	9F	000CF	PUSHAB	SHIFT	: 0941
			04	DD	000D5	PUSHL	#4	
00000000G	EF	00000000G	EF	9F	000D7	PUSHAB	PASS\$FV OUTPUT	
			03	FB	000DD	CALLS	#3,PASS\$WRITE_STRING	
			01	DD	000E4	PUSHL	#1	
	7E	00000000G	EF	9A	000E6	MOVZBL	TAB,-(SP)	
00000000G	EF	00000000G	EF	9F	000ED	PUSHAB	PASS\$FV OUTPUT	
			03	FB	000F3	CALLS	#3,PASS\$WRITE_CHAR	
			01	DD	000FA	PUSHL	#1	
	7E	00000000G	EF	9A	000FC	MOVZBL	TAB,-(SP)	
00000000G	EF	00000000G	EF	9F	00103	PUSHAB	PASS\$FV OUTPUT	
			03	FB	00109	CALLS	#3,PASS\$WRITE_CHAR	
		FFFFF103	EF	9F	00110	PUSHAB	C.ABB	
			19	DD	00116	PUSHL	#25	
00000000G	EF	00000000G	EF	9F	00118	PUSHAB	PASS\$FV OUTPUT	
			03	FB	0011E	CALLS	#3,PASS\$WRITE_STRING	
00000000G	EF	00000000G	EF	9F	00125	PUSHAB	PASS\$FV OUTPUT	
			01	FB	0012B	CALLS	#1,PASS\$WRITELN2	
			04	00132	5\$:	RET		: 0943

; Routine Size: 307 bytes, Routine Base: \$CODE + 00EFF

				0000	00000	.ENTRY	INPUT_ANALYSIS_FILE,^M<>	: 0988	
	5E		08	C2	00002	SUBL2	#8,SP		
		F8	AD	D4	00005	CLRL	-8(FP)		
03	00000000G	EF	9E	00008	MOVAB	PASS\$HANDLER,(FP)			
			00	E0	0000F	BBS	#0,EDITING,..+3	: 0995	
		0000V	31	00017	BRW	8\$			
03	00000000G	EF	00	E0	0001A	BBS	#0,ANALYSIS_SPECIFIED,..+3		
			0000V	31	00022	BRW	8\$		
	F8	AD	00000000G	EF	9E	00025	MOVAB	RMS_INPUT_COND_HANDLER,FP-8	: 1002
00V00000000G	EF		00	E0	0002D	BBS	#0,AUTO_TONE,4\$: 1007	
			EF	9F	00035	PUSHAB	SHIFT	: 1009	
			04	DD	0003B	PUSHL	#4		

Generated Code										
		F8	AD	D4	00005	CLRL	-8(FP)			
		6D	EF	9E	00008	MOVAB	PAS\$HANDLER,(FP)			
	F8	AD	EF	9E	0000F	MOVAB	CTRLZ_COND_HANDLER,FP-8	:	1170	
	00V	00000000G	EF	E0	00017	BBS	#0,AUTO_TUNE,3\$:	1172	
		00V	AF	9F	0001F	PUSHAB	3\$:	1174	
			19	DD	00022	PUSHL	#25			
		00000000G	EF	9F	00024	P'ISHAB	FDL_DEST			
	00000000G		03	FB	0002A	CALLS	#3,PAS\$CLOSE2			
	00000000G		00	FB	00031	CALLS	#0,POINT_AT_DEFINITION	:	1176	
	00000000G		01	90	00038	MOVB	#1,DEST_IS_TERMINAL	:	1177	
		00000000G	EF	94	0003F	CLRB	OPTIMIZING	:	1178	
		00000000G	FF	94	00045	CLRB	VISIBLE_QUESTION	:	1179	
		00000000G	EF	94	0004B	CLRB	TEMP_FUCL_PROMPT	:	1180	
	00000000G		EF	90	00051	MOVB	AUTO_TUNE_TAKE_DEFAULTS	:	1181	
		07	00000008G	EF	D1	0005C	CML	IDATA+8,#7	:	1186
			03	13	00063	BEQL	.+3			
			0000V	31	00065	BRW	17\$			
			8F	DF	00068	PUSHAL	#41	:	1193	
	00000000G		01	FB	0006E	CALLS	#1,QUERY			
08		00	000000A4G	EF	CF	00075	CASEL	IDATA+164,#0,#8	:	1195
			0000V		0007D	.DISPL	6\$			
			0000V		0007F	.DISPL	7\$			
			0000V		00081	.DISPL	14\$			
			0000V		00083	.DISPL	8\$			
			0000V		00085	.DISPL	9\$			
			0000V		00087	.DISPL	10\$			
			0000V		00089	.DISPL	11\$			
			0000V		0008B	.DISPL	12\$			
			0000V		0008D	.DISPL	13\$			
			00V	11	0008F	BRB	15\$			
	00000000G		00	FB	00091	CALLS	#0,ADD_FDL_LINE	:	1197	
			0000V	31	00098	BRW	20\$			
	00000000G		00	FB	0009B	CALLS	#0,DELETE_FDL_LINE	:	1198	
			00V	11	000A2	BRB	20\$			
	00000000G		00	FB	000A4	CALLS	#0,HELP_PROC	:	1199	
			00V	11	000AB	BRB	20\$			
	00000000G		CO	FB	000AD	CALLS	#0,INVOKE_SCRIPT	:	1200	
			00V	11	000B4	BRB	20\$			
	00000000G		00	FB	000B6	CALLS	#0,MODIFY_FDL_LINE	:	1201	
			00V	11	000BD	BRB	20\$			
			00000000G	EF	94	000BF	CLRB	EDITING	:	1202
			00V	11	000C5	BRB	20\$			
	00000000G		00	FB	000C7	CALLS	#0,SET_PROC	:	1203	
			00V	11	000CE	BRB	20\$			
	00000000G		00	FB	000D0	CALLS	#0,VIEW_DEF	:	1204	
			00V	11	000D7	BRB	20\$			
			00000000G	EF	94	000D9	CLRB	EDITING	:	1213
	00000000G		00	FB	000DF	CALLS	#0,CREATE_NEW_FDL	:	1214	
			00V	11	000E6	BRB	20\$			
			00V	11	000E8	BRB	20\$			
	00000108G		EF	D0	000EA	MOVL	IDATA+8,IDATA+264	:	1233	
			50	94	000F5	CLRB	RO	:	1235	
	00000100		8F	00000108G	EF	D1	000F7	CML	IDATA+264,#256	
			00V	1E	00102	BGEQU	19\$			
	00VFFFEEEDB		EF	00000108G	EF	E1	00104	BBC	IDATA+264,C.ABE,19\$	
			50	96	00110	INCB	RO			
	00000000G		EF	50	90	00112	MOVB	RO,ISAM_ORG		

COMMAND QUALIFIERS

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

/CHECK=(NOBOUNDS, NOCASE_SELECTORS, NOOVERFLOW, NOPOINTERS, NOSUBRANGE)

/DEBUG=(NOSYMBOLS, NOTRACEBACK)

/NOENVIRONMENT

/LIST= \$255\$DUA28:[EDF.LIS]EDFMAIN.LIS;1

/OBJECT= \$255\$DUA28:[EDF.OBJ]EDFMAIN.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	77	00:00.4	00:03.9
Source Analysis	1155	00:22.1	05:09.1
Source Listing	51	00:01.8	00:04.0
Tree Construction	253	00:01.1	00:02.8
Flow Analysis	13	00:00.3	00:00.8
Profit Analysis	51	00:00.5	00:01.7
Context Analysis	255	00:04.6	00:10.1
Name Packing	10	00:00.2	00:00.4
Code Selection	75	00:01.0	00:02.3
Final	140	00:03.9	00:10.5
TOTAL	2083	00:35.8	05:45.8

COMPILATION STATISTICS

CPU Time: 00:35.8 (2198 Lines/Minute)
Elapsed Time: 05:45.8
Page Faults: 2083
Compilation Complete

0127 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a large grid of approximately 100 small terminal windows, each containing text-based data. The windows are arranged in a regular pattern across the page. Several windows are clearly legible and contain the following text:

- EDFSOLMSG LIS**: Located in the upper right quadrant.
- EDFMAIN LIS**: Located in the middle right quadrant.
- EDFSHOW LIS**: Located in the middle right quadrant, below EDFMAIN LIS.
- EDFMSG LIS**: Located in the lower right quadrant.
- EDFGRF LIS**: Located in the lower left quadrant.

The text within the windows appears to be a mix of program headers, data lists, and status messages, typical of a multi-user operating system environment like VAX/VMS.