```
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     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  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
```

```
**FILE**ID**EDFUTIL
```

```
EEEEEEEEEE  DDDDDDDD    FFFFFFFFFF  UU      UU  TTTTTTTTTT    IIIIII    LL
EEEEEEEEEE  DDDDDDDD    FFFFFFFFFF  UU      UU  TTTTTTTTTT    IIIIII    LL
EE          DD      DD  FF          UU      UU      TT          II      LL
EE          DD      DD  FF          UU      UU      TT          II      LL
EE          DD      DD  FF          UU      UU      TT          II      LL
EE          DD      DD  FF          UU      UU      TT          II      LL
EEEEEEEE    DD      DD  FFFFFFFF    UU      UU      TT          II      LL
EEEEEEEE    DD      DD  FFFFFFFF    UU      UU      TT          II      LL
EE          DD      DD  FF          UU      UU      TT          II      LL
EE          DD      DD  FF          UU      UU      TT          II      LL
EE          DD      DD  FF          UU      UU      TT          II      LL             ....
EE          DD      DD  FF          UU      UU      TT          II      LL             ....
EEEEEEEEEE  DDDDDDDD    FF          UUUUUUUUUU      TT        IIIIII    LLLLLLLLLL     ....
EEEEEEEEEE  DDDDDDDD    FF          UUUUUUUUUU      TT        IIIIII    LLLLLLLLLL     ....

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
```

```
        [ 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:

                V03-012 RRB0005         Rowland R. Bradley       16 Jan 1984
                        Fix EDF$RESET_SCROLL to avoid overwriting the output
                        generated by a command procedure with SET VERIFY set.

                V03-011 RRB0004         Rowland R. Bradley       13 Jan 1984
                        Fix MAX_FACTOR to prevent division by zero.

                V03-010 KFH0010         Ken Henderson            10 Sep 1983
                        Support for named UICs.

                V03-009 KFH0009         Ken Henderson             8 Aug 1983
                        Changes for seperate compilation.

                V03-008 KFH0008         Ken Henderson            28 Jul 1983
                        Added CALC_REC_OVERHEAD and CALC_BUC_OVERHEAD
                        to centralize the arithmetic.
                        Fixed EDF$LINE_PARSED to not invert the
                        order of block comment lines.

                V03-007 KFH0007         Ken Henderson           26 Apr 1983
                        Fix NUMBER_INPUT to set INPUT_VALUE
                        and INPUT_NUMBER also. Modify
                        SCAN_DEFINITION routine to accept
                        FATAL parameter. Modify CURRENT_LT_TEST
                        and CURRENT_GT_TEST routines to
                        reverse precedence of SECONDARY
                        and SECNUM
                        - and make SEG_TYPE SECNUM = 7.

                V03-006 KFH0006         Ken Henderson           14 Apr 1983
                        Added support for JOURNAL_ENABLED.
                        Added MAX_FACTOR, DELETE_PRIMARY_SECTION
                        routines.

                V03-005 KFH0005         Ken Henderson           31 Jan 1983
                        Added XAB$C_BN8 and XAB$C_IN8 to
                        EDF$LINE_PARSED. And changed the
                        reference of FDL$TYPE to FDL3$TYPE.

                V03-004 KFH0004         Ken Henderson           11 Jan 1983
                        Modified EDF$RESET_SCROLL to say
                        "Created:" in reverse video.

                V03-003 KFH0003         Ken Henderson           8 Sept 1982
                        Modified reference to some variables
                        to fit with database reorganization.
                        Also, modified call to ASK_RETURN.

                V03-002 KFH0002         Ken Henderson           2 April 1982
                        Modified INSERT_IN_ORDER to not
                        start at DEF_HEAD if it was already
                        at the correct place.

                V03-001 KFH0001         Ken Henderson           23-Mar-1982
                        Modified EDF$RESET_SCROLL to not
                        reset the scrolling region unless
                        it has been set.

        -- }
        ENVIRONMENT ('LIB$:EDFUTIL'),

        INHERIT (

        'SYS$LIBRARY:STARLET',
        'SHRLIB$:FDLPARDEF',
        'LIB$:EDFSDLMSG',
        'LIB$:EDFSTRUCT',
        'LIB$:EDFCONST',
        'LIB$:EDFTYPE',
        'LIB$:EDFVAR',
        'LIB$:EDFEXTERN',
        'LIB$:EDFCHF'

        )]

        MODULE EDFUTIL (INPUT,OUTPUT);
```

```
{ ++

NUM_LEN -- Calculate the field width of an integer.

This routine will return the number of characters an integer will take up
when printed.

CALLING SEQUENCE:

field-width     := NUM_LEN (NUMBER);

INPUT PARAMETERS:

NUMBER

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

The field width

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] FUNCTION NUM_LEN (
                    NUMBER      : INTEGER
                    ) : INTEGER;

VAR
    TEST_VAR    : INTEGER;
    TEST_LEN    : INTEGER;

BEGIN

    IF NUMBER = 0 THEN

        { +
        Just plug a width of 1 if the number is 0.
        - }
        NUM_LEN                 := 1

    ELSE

    BEGIN

        { +
        Set the function value according to the magnitude of the number.
        - }
        TEST_VAR                := 1000000000;
        TEST_LEN                := 10;

        REPEAT

            IF ABS (NUMBER) < TEST_VAR THEN

                TEST_LEN        := TEST_LEN - 1;

            TEST_VAR            := TEST_VAR DIV 10;

        UNTIL ABS (NUMBER) >= TEST_VAR;

        { +
        Allow for a - sign if negative.
        - }
        IF NUMBER < 0 THEN

            TEST_LEN            := TEST_LEN + 1;

        { +
        Now stuff the function value.
        - }
        NUM_LEN                 := TEST_LEN;

    END;        { IF FALSE NUMBER = 0 }

END;    { NUM_LEN }
```

```
{ ++

MAX_FACTOR -- Produce a number that's a multiple of another.

This function will return the number that's a multiple of one of the
arguments, as long as it doesn't go over a maximum.

CALLING SEQUENCE:

NEWVALUE := MAX_FACTOR (BASE,VALUE,MAX);

INPUT PARAMETERS:

BASE
VALUE
MAX

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

The number (greater or equal to VALUE) that's a multiple of BASE,
unless that number would be greater than MAX - in which case it is MAX.

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
FUNCTION MAX_FACTOR (
                     BASE         : INTEGER;
                     VALUE        : INTEGER;
                     MAX          : INTEGER
                     ) : INTEGER;

VAR
    TEMP         : INTEGER;

BEGIN

    IF (VALUE < BASE) OR (BASE = 0) THEN

        TEMP    := BASE

    ELSE

    BEGIN

        TEMP    := VALUE DIV BASE;

        IF ((VALUE MOD BASE) <> 0) THEN

            TEMP        := TEMP + 1;

        TEMP    := TEMP * BASE;

    END;

    IF TEMP > MAX THEN

        TEMP    := MAX;

    MAX_FACTOR  := TEMP;

END;    { MAX_FACTOR }
```

```
{ ++

CALC_REC_OVERHEAD -- Do the arithmetic to figure out overheads.

This function will return the RECORD overhead for a given setup.

CALLING SEQUENCE:

RECORD_OVERHEAD := CALC_REC_OVERHEAD (INDEX_LEVEL);

INPUT PARAMETERS:

INDEX_LEVEL

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

The overhead, according to the RMS structure constants.

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
FUNCTION CALC_REC_OVERHEAD (
                    INDEX_LEVEL : INTEGER
                    ) : INTEGER;

VAR
    RECORD_OVERHEAD : INTEGER;

BEGIN

    RECORD_OVERHEAD     := 0;

    { +
    SIDR BUCKET
    - }
    IF (IDATA[EDF$K_ACTIVE_KEY] <> 0) AND (INDEX_LEVEL = 0) THEN

        RECORD_OVERHEAD := RECORD_OVERHEAD + IRC$C_SDROVHSZ3 + IRC$C_RRVOVHSZ3;

    { +
    ACCOUNT FOR KEY COMPRESSION
    - }
    IF (
    (BDATA[EDF$K_KEY_COMP_WANTED] AND (INDEX_LEVEL = 0))
    OR
    (BDATA[EDF$K_IDX_COMP_WANTED] AND (INDEX_LEVEL <> 0))
    ) THEN

        RECORD_OVERHEAD := RECORD_OVERHEAD + IRC$C_KEYCMPOVH;

    { +
    INDEX BUCKETS
    - }
    IF INDEX_LEVEL <> 0 THEN

        RECORD_OVERHEAD := RECORD_OVERHEAD + IRC$C_MAXVBNSZ;

    { +
    PRIMARY KEY DATA BUCKETS
    - }
    IF (IDATA[EDF$K_ACTIVE_KEY] = 0) AND (INDEX_LEVEL = 0) THEN

        BEGIN

        IF VARIABLE_RECORDS THEN

            RECORD_OVERHEAD     := RECORD_OVERHEAD + IRC$C_VAROVHSZ3

        ELSE

            RECORD_OVERHEAD     := RECORD_OVERHEAD + IRC$C_FIXOVHSZ3;

        IF BDATA[EDF$K_REC_COMP_WANTED] THEN

            RECORD_OVERHEAD     := RECORD_OVERHEAD + IRC$C_DATCMPOVH;

        IF BDATA[EDF$K_KEY_COMP_WANTED] OR BDATA[EDF$K_REC_COMP_WANTED] THEN

```

```
                    RECORD_OVERHEAD      := RECORD_OVERHEAD + IDATA[EDF$K_KEY_SIZE];

                END;

            CALC_REC_OVERHEAD    := RECORD_OVERHEAD;

        END;    { CALC_REC_OVERHEAD }
```

```
{ ++

CALC_BUC_OVERHEAD -- Do the arithmetic to figure out overheads.

This function will return the BUCKET overhead for a given setup.

CALLING SEQUENCE:

BUCKET_OVERHEAD := CALC_BUC_OVERHEAD (INDEX_LEVEL);

INPUT PARAMETERS:

INDEX_LEVEL

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

The overhead, according to the RMS structure constants.

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
FUNCTION CALC_BUC_OVERHEAD (
                INDEX_LEVEL : INTEGER
                ) : INTEGER;

BEGIN

    IF INDEX_LEVEL = 0 THEN

        CALC_BUC_OVERHEAD       := BKT$C_OVERHDSZ + BKT$C_DATBKTOVH

    ELSE

        CALC_BUC_OVERHEAD       := BKT$C_OVERHDSZ + BKT$C_ENDOVHD;

END;    { CALC_BUC_OVERHEAD }
```

```
{ ++

EDF$RESET_SCROLL -- Reset an ANSI terminal's scroll region.

This routine will put the scroll region back to full screen.
It also clears graphics mode.
It is a Global routine, which is called by the exit handler as well.

CALLING SEQUENCE:

EDF$RESET_SCROLL;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

LINE_ONE
LINES_PER_PAGE

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

SYS$OUTPUT: - the scroll region is reset, and possibly graphics mode reset

ROUTINES CALLED:

LIB$SET_SCROLL

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS,GLOBAL] PROCEDURE EDF$RESET_SCROLL;

BEGIN

    IF NOT AUTO_TUNE THEN

    BEGIN

        { +
        Clear graphics mode if this is a Regis device.
        - }
        IF REGIS THEN

        BEGIN

            CHFFLAGS    := 0;
            WRITEV (OUT_LINE,''(27)'\');
            LIB$PUT_LINE (OUT_LINE,ONE,CHFFLAGS);

        END;

        { +
        Now make the scroll region from top to bottom - if it was ever set.
        - }
        IF SCROLLING_SET THEN

            LIB$SET_SCROLL (LINE_ONE,LINES_PER_PAGE);

    END;        { IF NOT AUTO_TUNE }

    { +
    Announce that the file has been created.
    - }
    IF (
    (FILE_CREATED)
    AND
    (RES_OUTPUT_FILENAME_DESC.DSC$W_LENGTH > 0)
    ) THEN

    BEGIN

        CHFFLAGS        := SCR$M_REVERSE;
        WRITEV (OUT_LINE,CRLF,
RES_OUTPUT_FILENAME_DESC.DSC$A_POINTER^:RES_OUTPUT_FILENAME_DESC.DSC$W_LENGTH,
'  ',LINES_SHOWN:NUM_LEN(LINES_SHOWN),' lines');
        LIB$PUT_LINE (OUT_LINE,ONE,CHFFLAGS);

    END;

END;    { EDF$RESET_SCROLL }
```

```
{ ++

CLEAR -- Clear a designated area of the screen.

This routine clears a specific area on the screen and leaves the cursor there.
It bypasses screwing up non-CRT terminals.

CALLING SEQUENCE:

CLEAR (DESTINATION);

INPUT PARAMETERS:

DESTINATION

IMPLICIT INPUTS:

PROMPT_LINE
LINE_ONE

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

SYS$OUTPUT:

ROUTINES CALLED:

LIB$ERASE_PAGE
LIB$ERASE_LINE

ROUTINE VALUE:

none

SIGNALS:


SIDE EFFECTS:

The selected lines on the screen are cleared (unless hardcopy).

-- }
```

```
PROCEDURE CLEAR (
                DESTINATION : INTEGER
                );

BEGIN

    { +
    All this stuff affects only video terminals.
    - }
    IF (
    (VIDEO_TERMINAL)
    AND
    (NOT AUTO_TUNE)
    ) THEN

    BEGIN

        CASE DESTINATION OF

            SCREEN :

                BEGIN

                    { +
                    The following sequence of junk to the screen
                    is overkill to make sure the titles don't
                    jump around. (interaction of Pascal I/O and
                    screen package I/O...)
                    - }
                    IF REGIS THEN

                        WRITELN (''(27)'Pp;S(E);'(27)'\');

                    LIB$ERASE_PAGE (LINE_ONE,COL_ONE);
                    WRITELN (' ');
                    LIB$SET_CURSOR (LINE_ONE,COL_ONE);

                END;    { SCREEN }

            LOWER_AREA :

            BEGIN

                IF REGIS THEN

                BEGIN

                    WRITELN (
                    ''(27)'PpP[27,320];V(W(I0,S1,E,S[,479]))[+767];'(27)'\');

                    LIB$SET_CURSOR (PROMPT_LINE,COL_ONE);

                END

                ELSE

                    LIB$ERASE_PAGE (LOWER_LINE,COL_ONE);

                    END;

                    IF_FULL_PROMPT :    IF FULL_PROMPT OR TEMP_FULL_PROMPT THEN

                        BEGIN

                            IF TEMP_FULL_PROMPT THEN

                                LIB$WAIT (1.3);

                            { +
                            The following sequence of junk to the screen
                            is overkill to make sure the titles don't
                            jump around. (interaction of Pascal I/O and
                            screen package I/O...)
                            - }

                            IF REGIS THEN

                                WRITELN (''(27)'Pp;S(E);'(27)'\');

                            LIB$ERASE_PAGE (LINE_ONE,COL_ONE);
                            WRITELN (' ');
                            LIB$SET_CURSOR (LINE_ONE,COL_ONE);

                        END;    { IF_FULL_PROMPT }

                    PAUSE :             QUERY (EDF$K_RETURN);

                OTHERWISE

                    { NULL-STATMENT } ;

                END;            { CASE }

            END;        { IF VIDEO_TERMINAL AND NOT AUTO_TUNE }

        END;    { CLEAR }
```

```
{ ++

CVT_QUAD_DESC -- Routine to convert a quadword to a descriptor.

This routine will take 2 longword arguments and stuff them into a descriptor.

CALLING SEQUENCE:

DESCRIPTOR_VAR  := CVT_QUAD_DESC (LONG1,LONG2);

INPUT PARAMETERS:

LONG1
LONG2

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

DESCRIPTOR_VAR

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] FUNCTION CVT_QUAD_DESC (LONG1,LONG2 : LONG) : DESCRIPTOR;

BEGIN

    WITH QUAD_DESC DO

    BEGIN

        { +
        Select the quadword type variant and stuff it.
        - }
        QWHICH                  := QWORD;
        TWOLONG.L1              := LONG1;
        TWOLONG.L2              := LONG2;

        { +
        Now select the descriptor type variant and get it.
        - }
        QWHICH                  := DWORD;
        CVT_QUAD_DESC           := DSC;

    END;

END;    { CVT_QUAD_DESC }
```

```
{ ++

SCAN_DEFINITION -- Search for area and key primaries in the current definition
and log them.

CALLING SEQUENCE:

SCAN_DEFINITION (FATAL);

INPUT PARAMETERS:

FATAL

IMPLICIT INPUTS:

DEF_CURRENT

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

LOW_AREA
HIGH_AREA
LOW_KEY
HIGH_KEY
FOUND_0
FOUND_AREA
FOUND_KEY

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
PROCEDURE SCAN_DEFINITION (FATAL : BOOLEAN);

BEGIN

    { +
    Find out the range of existing keys (assume contiguous).
    - }
    DEF_CURRENT          := DEF_HEAD;
    FOUND_0              := FALSE;
    FOUND_AREA           := FALSE;
    FOUND_KEY            := FALSE;
    LOW_AREA             := 0;
    HIGH_AREA            := 0;
    LOW_KEY              := 0;
    HIGH_KEY             := 0;

    REPEAT

        WITH DEF_CURRENT^ DO

        BEGIN

            IF (
            (OBJECT_TYPE = PRI)
            AND
            (PRIMARY = KEY)
            ) THEN

            BEGIN

                IF PRINUM = 0 THEN

                    FOUND_0      := TRUE;

                FOUND_KEY        := TRUE;

                IF PRINUM < LOW_KEY THEN

                    LOW_KEY      := PRINUM;

                IF PRINUM > HIGH_KEY THEN

                    HIGH_KEY     := PRINUM;

            END;

            IF (
            (OBJECT_TYPE = PRI)
            AND
            (PRIMARY = AREA)
            ) THEN

            BEGIN

                FOUND_AREA       := TRUE;

                IF PRINUM < LOW_AREA THEN
```

```

                                      LOW_AREA     := PRINUM;

                                  IF PRINUM > HIGH_AREA THEN

                                      HIGH_AREA    := PRINUM;

                              END;

                          END;    { WITH DEF_CURRENT^ DO }

                          DEF_CURRENT      := DEF_CURRENT^.FORE;

                      UNTIL DEF_CURRENT = NIL;

                      IF (
                      ((FATAL) OR (HIGH_KEY <> 0))
                      AND
                      (NOT FOUND_0)
                      ) THEN

                      BEGIN

                          WRITELN (SHIFT,ANSI_REVERSE,
                          ' There is no Primary Key in the Current Definition. ',
                          ANSI_RESET);

                          IF AUTO_TUNE THEN

                              LIB$STOP (EDF$_INSFANL,0,0,0)

                          ELSE

                          BEGIN

                              LIB$WAIT (3,0);
                              LIB$SIGNAL (EDF$_CTRLZ,0,0,0);

                          END;

                      END;

                  END;    { SCAN_DEFINITION }
```

```
{ ++

PARSE_INPUT -- Routine to parse input string.

This routine will look at the chosen LIB$TPARSE table.

CALLING SEQUENCE:

PARSE_INPUT (KEY_TABLE_PTR,STATE_TABLE_PTR,DEFAULT_OK,DEFAULT_VALUE);

INPUT PARAMETERS:

KEY_TABLE_PTR
STATE_TABLE_PTR

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

INPUT_VALUE

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
PROCEDURE PARSE_INPUT (
                        KEY_TABLE         : INTEGER;
                        STATE_TABLE       : INTEGER;
                        DEFAULT_OK        : BOOLEAN;
                        DEFAULT_VALUE     : INTEGER
                        );

BEGIN

    { +
    Get the input from the terminal.
    - }
    INPUT_DESC  := NULL_STRING;

    { +
    If auto answers are enabled and this question has a default - use it.
    - }
    IF (
    (
    (TAKE_DEFAULTS)
    AND
    (IDATA[EDF$K_RESPONSES] = EDF$K_AUTO)
    AND
    (DEFAULT_OK)
    AND
    (NOT (QTAB_OFFSET = EDF$K_RETURN))
    )
    OR
    (AUTO_TUNE)
    ) THEN

    BEGIN

        IF NOT AUTO_TUNE THEN

        BEGIN

            WRITELN (CRLF);
            LIB$WAIT (0.7);

        END;

    END

    ELSE

    BEGIN

        IF EOF (INPUT) THEN

        BEGIN

            RESET (INPUT);
            LIB$SIGNAL (EDF$_CTRLZ,0,0,0);

        END;

```

```
            READLN (INPUT_STRING);
            WRITELN (CRLF);
            STR$TRIM (INPUT_DESC,INPUT_STRING);
            STR$UPCASE (INPUT_DESC,INPUT_DESC);

        END;

        { +
        If we're journaling our input, save a copy of it to the
        journal file.
        - }
        IF JOURNAL_ENABLED THEN

            IF INPUT_DESC.DSC$W_LENGTH > 0 THEN

                WRITELN (
                    JOURNAL_FILE,
                    INPUT_DESC.DSC$A_POINTER^:INPUT_DESC.DSC$W_LENGTH
                    )

            ELSE

                WRITELN (JOURNAL_FILE);

        { +
        See if the answer was defaulted, and if it's allowed to be.
        - }
        IF INPUT_DESC.DSC$W_LENGTH = 0 THEN

            IF DEFAULT_OK THEN

                INPUT_VALUE          := DEFAULT_VALUE

            ELSE

            BEGIN

                LIB$SIGNAL (EDF$_NODEFAULT,0,0,0);

            END

        ELSE

        BEGIN

            { +
            See if it's valid and get it's value.
            - }
            PARAM_BLOCK.TPA$L_STRINGPTR      := INPUT_DESC.DSC$A_POINTER::UNSIGNED;
            PARAM_BLOCK.TPA$L_STRINGCNT      := INPUT_DESC.DSC$W_LENGTH;

            ISTATUS := LIB$TPARSE (
                                  PARAM_BLOCK,
                                  STATE_TABLE,
                                  KEY_TABLE
                                  );

                INPUT_VALUE     := PARAM_BLOCK.TPA$L_PARAM::LONG;
                INPUT_NUMBER    := PARAM_BLOCK.TPA$L_NUMBER::LONG;

                { +
                Even Istatus (low bit clear) means failure.
                - }
                IF NOT ODD (ISTATUS) THEN

                BEGIN

                    IF PARAM_BLOCK.TPA$V_AMBIG THEN

                        LIB$SIGNAL (EDF$_AMBIG,0,0,0)

                    ELSE

                        LIB$SIGNAL (EDF$_BADSYNTAX,0,0,0);

                END;

            END;    { IF NOT INPUT_DESC.DSC$W_LENGTH = 0 }

        END;    { PARSE_INPUT }
```

```
{ ++

NUMBER_INPUT -- Routine to get a number from the input string.

This routine will return the integer typed.

CALLING SEQUENCE:

NUMBER_INPUT (NUM_VALUE,DEFAULT_OK,DEFAULT_VALUE);

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

NUM_VALUE

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
        PROCEDURE NUMBER_INPUT (
                                VAR NUM_VALUE     : INTEGER;
                                DEFAULT_OK        : BOOLEAN;
                                DEFAULT_VALUE     : INTEGER
                                );

        BEGIN

            { +
            Get the input from the terminal.
            - }
            INPUT_DESC  := NULL_STRING;

            { +
            If auto answers are enabled and this question has a default - use it.
            - }
            IF (
            (
            (TAKE_DEFAULTS)
            AND
            (IDATA[EDF$K_RESPONSES] = EDF$K_AUTO)
            AND
            (DEFAULT_OK)
            )
            OR
            (AUTO_TUNE)
            ) THEN

            BEGIN

                IF NOT AUTO_TUNE THEN

                BEGIN

                    WRITELN (CRLF);
                    LIB$WAIT (0.7);

                END;

            END

            ELSE

            BEGIN

                IF EOF (INPUT) THEN

                BEGIN

                    RESET (INPUT);
                    LIB$SIGNAL (EDF$_CTRLZ,0,0,0);

                END;

                READLN (INPUT_STRING);
                WRITELN (CRLF);
                STR$TRIM (INPUT_DESC,INPUT_STRING);
                STR$UPCASE (INPUT_DESC,INPUT_DESC);
                PARAM_BLOCK.TPA$L_TOKENPTR      := INPUT_DESC.DSC$A_POINTER::UNSIGNED;
                PARAM_BLOCK.TPA$L_TOKENCNT      := INPUT_DESC.DSC$W_LENGTH;

            END;

            { +
            If we're journaling our input, save a copy of it to the
            journal file.
            - }
            IF JOURNAL_ENABLED THEN

                IF INPUT_DESC.DSC$W_LENGTH > 0 THEN

                    WRITELN (
                        JOURNAL_FILE,
                        INPUT_DESC.DSC$A_POINTER^:INPUT_DESC.DSC$W_LENGTH
                        )

                ELSE

                    WRITELN (JOURNAL_FILE);

            { +
            See if the answer was defaulted, and if it's allowed to be.
            - }
            IF INPUT_DESC.DSC$W_LENGTH = 0 THEN

                IF DEFAULT_OK THEN

                    NUM_VALUE           := DEFAULT_VALUE

                ELSE

                BEGIN

                    LIB$SIGNAL (EDF$_NODEFAULT,0,0,0);

                END

            ELSE

            BEGIN

                { +
                Convert it to an integer.
                - }
                ISTATUS := OTS$CVT_TI_L (INPUT_DESC,NUM_VALUE);

                { +
                Even Istatus (low bit clear) means failure.
                - }
                IF NOT ODD (ISTATUS) THEN

                BEGIN

                    LIB$SIGNAL (EDF$_BADSYNTAX,0,0,0);

                END;

            END;    { IF NOT INPUT_DESC.DSC$W_LENGTH = 0 }

            INPUT_VALUE         := NUM_VALUE;
            INPUT_NUMBER        := NUM_VALUE;

        END;    { NUMBER_INPUT }
```

```
{ ++

MAKE_SCRATCH -- Create a new peice of dynamic memory and init it.

This routine creates a new Line_object, and inits its various fields.

CALLING SEQUENCE:

MAKE_SCRATCH;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

LINE_OBJECT_TEMPLATE

OUTPUT PARAMETERS:

DEF_SCRATCH

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] PROCEDURE MAKE_SCRATCH;

BEGIN

    { +
    Allocate some dynamic memory.
    - }
    NEW (DEF_SCRATCH);

    { +
    Copy over the template.
    - }
    DEF_SCRATCH^          := LINE_OBJECT_TEMPLATE;

END;    { MAKE_SCRATCH }
```

```
{ ++

CURRENT_GT_TEST -- Compare def_current with test.

This function has the boolean value of the whether or not DEF_CURRENT is
greater than TEST.

CALLING SEQUENCE:

test-val        := CURRENT_GT_TEST (TEST,SCOPE);

INPUT PARAMETERS:

SCOPE

IMPLICIT INPUTS:

DEF_CURRENT

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

True if DEF_CURRENT > TEST, false if not.

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] FUNCTION CURRENT_GT_TEST (
                                TEST                : LINE_OBJECT;
                                EXACT_COMPARISON    : BOOLEAN
                                ) : BOOLEAN;

BEGIN

    CURRENT_GT_TEST     := FALSE;

    { +
    Just do a boolean assignment.
    - }
    IF EXACT_COMPARISON THEN

    BEGIN

        IF
        (PRI_SEQ[DEF_CURRENT^.PRIMARY] > PRI_SEQ[TEST.PRIMARY])
        THEN

            CURRENT_GT_TEST     := TRUE;

        IF
        (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
        AND
        (DEF_CURRENT^.PRINUM > TEST.PRINUM)
        THEN

            CURRENT_GT_TEST     := TRUE;

        IF
        (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
        AND
        (DEF_CURRENT^.PRINUM = TEST.PRINUM)
        AND
        (DEF_CURRENT^.SECNUM > TEST.SECNUM)
        THEN

            CURRENT_GT_TEST     := TRUE;

        IF
        (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
        AND
        (DEF_CURRENT^.PRINUM = TEST.PRINUM)
        AND
        (DEF_CURRENT^.SECNUM = TEST.SECNUM)
        AND
        (DEF_CURRENT^.SECONDARY > TEST.SECONDARY)
        THEN

            CURRENT_GT_TEST     := TRUE;

    END

    ELSE

    BEGIN
```

```

                    IF(
                    PRI_SEQ[DEF_CURRENT^.PRIMARY] > PRI_SEQ[TEST.PRIMARY])
                    THEN

                        CURRENT_GT_TEST      := TRUE;

                    IF
                    (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
                    AND
                    (DEF_CURRENT^.PRINUM > TEST.PRINUM)
                    THEN

                        CURRENT_GT_TEST      := TRUE;

                END;

            END;    { CURRENT_GT_TEST }
```

```
{ ++

CURRENT_LT_TEST -- Compare def_current and test.

This function has the boolean value of the whether or not TEST greater than
DEF_CURRENT.

CALLING SEQUENCE:

test-val        := CURRENT_LT_TEST (TEST,SCOPE);

INPUT PARAMETERS:

TEST

IMPLICIT INPUTS:

DEF_CURRENT

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

True if DEF_CURRENT < TEST, false if not.

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
        [ASYNCHRONOUS] FUNCTION CURRENT_LT_TEST (
                                TEST                : LINE_OBJECT;
                                EXACT_COMPARISON    : BOOLEAN
                                ) : BOOLEAN;

        BEGIN

            CURRENT_LT_TEST     := FALSE;

            { +
            Just do a boolean assignment.
            - }
            IF EXACT_COMPARISON THEN

            BEGIN

                IF(
                PRI_SEQ[DEF_CURRENT^.PRIMARY] < PRI_SEQ[TEST.PRIMARY])
                THEN

                    CURRENT_LT_TEST     := TRUE;

                IF
                (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
                AND
                (DEF_CURRENT^.PRINUM < TEST.PRINUM)
                THEN

                    CURRENT_LT_TEST     := TRUE;

                IF
                (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
                AND
                (DEF_CURRENT^.PRINUM = TEST.PRINUM)
                AND
                (DEF_CURRENT^.SECNUM < TEST.SECNUM)
                THEN

                    CURRENT_LT_TEST     := TRUE;

                IF
                (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
                AND
                (DEF_CURRENT^.PRINUM = TEST.PRINUM)
                AND
                (DEF_CURRENT^.SECNUM = TEST.SECNUM)
                AND
                (DEF_CURRENT^.SECONDARY < TEST.SECONDARY)
                THEN

                    CURRENT_LT_TEST     := TRUE;

            END

            ELSE

            BEGIN
```

```

                IF(
                PRI_SEQ[DEF_CURRENT^.PRIMARY] < PRI_SEQ[TEST.PRIMARY])
                THEN

                    CURRENT_LT_TEST     := TRUE;

                IF
                (PRI_SEQ[DEF_CURRENT^.PRIMARY] = PRI_SEQ[TEST.PRIMARY])
                AND
                (DEF_CURRENT^.PRINUM < TEST.PRINUM)
                THEN

                    CURRENT_LT_TEST     := TRUE;

            END;

        END;    { CURRENT_LT_TEST }
        { ++

        CURRENT_EQ_TEST -- Compare def_current and test.

        This function has the boolean value of the whether or not TEST is the
        same as DEF_CURRENT.

        CALLING SEQUENCE:

        test-val        := CURRENT_EQ_TEST (TEST,SCOPE);

        INPUT PARAMETERS:

        SCOPE

        IMPLICIT INPUTS:

        DEF_CURRENT

        OUTPUT PARAMETERS:

        none

        IMPLICIT OUTPUTS:

        none

        ROUTINES CALLED:

        none

        ROUTINE VALUE:

        True if DEF_CURRENT = TEST, false if not.

        SIGNALS:

        none

        SIDE EFFECTS:

        none

        -- }
```

```
[ASYNCHRONOUS] FUNCTION CURRENT_EQ_TEST (
                        TEST                : LINE_OBJECT;
                        EXACT_COMPARISON    : BOOLEAN
                        ) : BOOLEAN;

BEGIN

    { +
    Just do a boolean assignment.
    - }
    IF EXACT_COMPARISON THEN

        CURRENT_EQ_TEST := (

        (TEST.OBJECT_TYPE = DEF_CURRENT^.OBJECT_TYPE)
        AND
        (TEST.PRIMARY = DEF_CURRENT^.PRIMARY)
        AND
        (TEST.PRINUM = DEF_CURRENT^.PRINUM)
        AND
        (TEST.SECONDARY = DEF_CURRENT^.SECONDARY)
        AND
        (TEST.SECNUM = DEF_CURRENT^.SECNUM)

        )

    ELSE

        CURRENT_EQ_TEST := (

        (TEST.PRIMARY = DEF_CURRENT^.PRIMARY)
        AND
        (TEST.PRINUM = DEF_CURRENT^.PRINUM)

        );

END;    { CURRENT_EQ_TEST }
```

```
{ ++

INSERT_BEFORE_CURRENT -- Link the DEF_SCRATCH line_object into the list.

This routine adds DEF_SCRATCH into the list just before DEF_CURRENT.

CALLING SEQUENCE:

INSERT_BEFORE_CURRENT;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

DEF_SCRATCH
DEF_CURRENT
DEF_HEAD

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
        [ASYNCHRONOUS] PROCEDURE INSERT_BEFORE_CURRENT;

        BEGIN

            { +
            Make it the new head, if we're adding it before the old head.
            - }
            IF DEF_CURRENT = DEF_HEAD THEN

                DEF_HEAD            := DEF_SCRATCH;

            { +
            Update the fore and back pointers.
            - }
            DEF_PRED            := DEF_CURRENT^.BACK;
            DEF_SCRATCH^.FORE   := DEF_CURRENT;
            DEF_SCRATCH^.BACK   := DEF_PRED;

            IF DEF_PRED <> NIL THEN

                DEF_PRED^.FORE  := DEF_SCRATCH;

            DEF_CURRENT^.BACK   := DEF_SCRATCH;

            { +
            Leave looking at the just inserted line_object.
            - }
            DEF_CURRENT         := DEF_SCRATCH;

        END;    { INSERT_BEFORE_CURRENT }
```

```
{ ++

INSERT_AT_CURRENT -- Link the DEF_SCRATCH line_object into the list.

This routine adds DEF_SCRATCH into the list at DEF_CURRENT.

CALLING SEQUENCE:

INSERT_AT_CURRENT;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

DEF_SCRATCH
DEF_CURRENT
DEF_HEAD

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
        [ASYNCHRONOUS] PROCEDURE INSERT_AT_CURRENT;

        BEGIN

            { +
            Make new head or tail, if we're replacing this one.
            - }
            IF DEF_CURRENT = DEF_HEAD THEN

                DEF_HEAD            := DEF_SCRATCH;

            IF DEF_CURRENT = DEF_TAIL THEN

                DEF_TAIL            := DEF_SCRATCH;

            { +
            Substitute the links to def_current with links to def_scratch.
            - }
            DEF_PRED            := DEF_CURRENT^.BACK;
            DEF_SUCC            := DEF_CURRENT^.FORE;
            DEF_SCRATCH^.FORE   := DEF_CURRENT^.FORE;
            DEF_SCRATCH^.BACK   := DEF_CURRENT^.BACK;

            IF DEF_PRED <> NIL THEN

                DEF_PRED^.FORE  := DEF_SCRATCH;

            IF DEF_SUCC <> NIL THEN

                DEF_SUCC^.BACK  := DEF_SCRATCH;

            { +
            Get rid of the old def_current, and point def_current to the new king.
            - }
            DISPOSE (DEF_CURRENT);

            DEF_CURRENT         := DEF_SCRATCH;

        END;    { INSERT_AT_CURRENT }
```

```
{ ++

INSERT_AFTER_CURRENT -- Link the DEF_SCRATCH line_object into the list.

This routine adds DEF_SCRATCH to the list after DEF_CURRENT.

CALLING SEQUENCE:

INSERT_AFTER_CURRENT;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

DEF_CURRENT
DEF_SCRATCH
DEF_TAIL

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] PROCEDURE INSERT_AFTER_CURRENT;

BEGIN

    { +
    Make it the new tail if we're adding it after the old tail.
    - }
    IF DEF_CURRENT = DEF_TAIL THEN

        DEF_TAIL        := DEF_SCRATCH;

    { +
    Update the fore and back pointers.
    - }
    DEF_SUCC            := DEF_CURRENT^.FORE;
    DEF_SCRATCH^.FORE   := DEF_SUCC;
    DEF_SCRATCH^.BACK   := DEF_CURRENT;
    DEF_CURRENT^.FORE   := DEF_SCRATCH;

    IF DEF_SUCC <> NIL THEN

        DEF_SUCC^.BACK  := DEF_SCRATCH;

    { +
    Leave looking at the just inserted line_object.
    - }
    DEF_CURRENT         := DEF_SCRATCH;

END;    { INSERT_AFTER_CURRENT }
```

```
{ ++

INCR_CURRENT -- Bump DEF_CURRENT pointer one.

This routine makes DEF_CURRENT point to the next line_object in the list,
as long as it won't fall off the end.

CALLING SEQUENCE:

INCR_CURRENT;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

DEF_CURRENT

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

LIB$SIGNAL

ROUTINE VALUE:

none

SIGNALS:


SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] PROCEDURE INCR_CURRENT;

BEGIN

    { +
    .FORE points to the next line_object in the list.
    - }
    IF DEF_CURRENT <> NIL THEN

        DEF_CURRENT      := DEF_CURRENT^.FORE;

END;    { INCR_CURRENT }
```

```
{ ++

DECR_CURRENT -- Bump back the DEF_CURRENT pointer one.

This routine points DEF_CURRENT to the previous line_object in the list,
as long as it won't run off the end.

CALLING SEQUENCE:

DECR_CURRENT;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

DEF_CURRENT

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

LIB$SIGNAL

ROUTINE VALUE:

none

SIGNALS:


SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS] PROCEDURE DECR_CURRENT;

BEGIN

    { +
    .BACK points to the previous line_object in the list.
    - }
    IF DEF_CURRENT <> NIL THEN

        DEF_CURRENT      := DEF_CURRENT^.BACK;

END;    { DECR_CURRENT }
```

```
{ ++

NEW_IDENT_LINE -- Make a new Ident and stuff it into the definition.

This routine gets the date and time from the system and makes a new Ident
Line_object, and puts it into the definition linked list.

IT ASSUMES THAT THE LIST IS COMPLETELY EMPTY!!!

CALLING SEQUENCE:

NEW_IDENT_LINE;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

DEF_SCRATCH
IDENT_STRING

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

DEF_SCRATCH
DEF_HEAD
DEF_TAIL
DEF_CURRENT

ROUTINES CALLED:

LIB$DATE_TIME
MAKE_SCRATCH

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
PROCEDURE NEW_IDENT_LINE;

VAR
    DATE_STRING : STRING20;
    I           : INTEGER;

BEGIN

    { +
    Create a place for the Ident to go.
    - }
    MAKE_SCRATCH;

    { +
    Get system date and time to put into IDENT line.
    - }
    LIB$DATE_TIME (DATE_STRING);

    { +
    Now, copy it into the ident string.
    - }
    FOR I := 1 TO 20 DO

        IDENT_STRING[I]         := DATE_STRING[I];

    { +
    Put an IDENT primary at the head of the linked list.
    - }
    WITH DEF_SCRATCH^ DO

    BEGIN

        TEMP_DESCRIPTOR                     := NULL_STRING;
        NEW (TEMP_DESCRIPTOR.DSC$A_POINTER);

        TEMP_DESCRIPTOR.DSC$W_LENGTH        := IDENT_STRING_LENGTH;
        OBJECT_TYPE                         := PRI;
        PRIMARY                             := IDENT;

        FOR I := 1 TO IDENT_STRING_LENGTH DO

            TEMP_DESCRIPTOR.DSC$A_POINTER^[I]   := IDENT_STRING[I];

        LIB$SCOPY_DXDX (TEMP_DESCRIPTOR,STRING);
        DISPOSE (TEMP_DESCRIPTOR.DSC$A_POINTER);

    END;        { WITH DEF_SCRATCH^ }

    { +
    Make the just created line_object the head (and only) one
    - }
    DEF_CURRENT         := DEF_SCRATCH;
    DEF_HEAD            := DEF_SCRATCH;
    DEF_TAIL            := DEF_SCRATCH;

END;    { NEW_IDENT_LINE }
```

```
{ ++

DELETE_CURRENT -- Unlink DEF_CURRENT from the list and kill it.

This routine removes the line_object pointed to by DEF_CURRENT from the list.

CALLING SEQUENCE:

DELETE_CURRENT;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

DEF_CURRENT
DEF_TAIL
DEF_HEAD

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

DEF_CURRENT
DEF_TAIL
DEF_HEAD

ROUTINES CALLED:

EXTRACT_CURRENT

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
          PROCEDURE DELETE_CURRENT;

          BEGIN

              IF DEF_CURRENT^.PRIMARY = TITLE THEN

              BEGIN

                  { +
                  TITLE is always the very 1st line_object in the list.
                  - }
                  IF DEF_CURRENT^.FORE = NIL THEN

                  BEGIN

                      DISPOSE (DEF_CURRENT);
                      NEW_IDENT_LINE;

                  END       { IF TRUE DEF_CURRENT^.FORE = NIL }

                  ELSE

                  BEGIN

                      DEF_HEAD              := DEF_CURRENT^.FORE;
                      DEF_HEAD^.BACK        := NIL;
                      DISPOSE (DEF_CURRENT);
                      DEF_CURRENT           := DEF_HEAD;

                  END;      { IF FALSE DEF_CURRENT^.FORE = NIL }

              END           { IF TRUE DEF_CURRENT^.PRIMARY = TITLE }

              ELSE

              BEGIN

              { +
              Make new tail, if we're deleting old tail.
              - }
              IF (DEF_CURRENT <> NIL) AND (DEF_CURRENT = DEF_TAIL) THEN

                  DEF_TAIL         := DEF_CURRENT^.BACK;

              { +
              Make new head, if we're deleting old head.
              - }
              IF (DEF_CURRENT <> NIL) AND (DEF_CURRENT = DEF_HEAD) THEN

                  DEF_HEAD         := DEF_CURRENT^.FORE;

              { +
              Update fore and back pointers.
              - }
              DEF_PRED             := DEF_CURRENT^.BACK;
              DEF_SUCC             := DEF_CURRENT^.FORE;
```

```
        IF DEF_PRED <> NIL THEN

            DEF_PRED^.FORE  := DEF_SUCC;

        IF DEF_SUCC <> NIL THEN

            DEF_SUCC^.BACK  := DEF_PRED;

        WITH DEF_CURRENT^ DO

        BEGIN

            IF STRING.DSC$W_LENGTH > 0 THEN

                STR$FREE1_DX (STRING);

            IF COMMENT.DSC$W_LENGTH > 0 THEN

                STR$FREE1_DX (COMMENT);

        END;

        DISPOSE (DEF_CURRENT);

        IF DEF_SUCC <> NIL THEN

            DEF_CURRENT     := DEF_SUCC

        ELSE IF DEF_PRED <> NIL THEN

            DEF_CURRENT     := DEF_PRED

        ELSE

            NEW_IDENT_LINE;

        END;        { IF FALSE DEF_CURRENT^.PRIMARY = TITLE }

    END;    { DELETE_CURRENT }
```

```
{ ++

DELETE_PRIMARY_SECTION -- Get rid of a whole primary section.

This routine removes all the line_objects of particular primary from the
definition linked list.

CALLING SEQUENCE:

DELETE_PRIMARY_SECTION (PRIMARY,PRINUM);

INPUT PARAMETERS:

PRIMARY
PRINUM

IMPLICIT INPUTS:

DEF_CURRENT
DEF_TAIL
DEF_HEAD

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

DEF_CURRENT
DEF_TAIL
DEF_HEAD

ROUTINES CALLED:

EXTRACT_CURRENT

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
        PROCEDURE DELETE_PRIMARY_SECTION (WHICHPRIMARY : PRIMARY_TYPE;
                                          WHICHPRINUM : INTEGER);

        VAR
            DOING         : BOOLEAN;
            DONE          : BOOLEAN;

        BEGIN

            DEF_CURRENT := DEF_HEAD;
            DOING       := FALSE;
            DONE        := FALSE;

            { +
            Cycle until we've deleted the section or we're at the end of the list.
            - }
            REPEAT

                { +
                If this is the start of the right primary, flag it.
                - }
                IF (
                (DEF_CURRENT^.OBJECT_TYPE = PRI)
                AND
                (DEF_CURRENT^.PRIMARY = WHICHPRIMARY)
                AND
                (DEF_CURRENT^.PRINUM = WHICHPRINUM)
                ) THEN

                    DOING       := TRUE;

                { +
                If we're in the right primary, delete the sucker!
                - }
                IF DOING THEN

                    DELETE_CURRENT

                ELSE

                    { +
                    Move on to the next line_object in the list.
                    - }
                    INCR_CURRENT;

                { +
                If we're not already off the end, see if this is still the
                right primary. If not, flag that we're done.
                - }
                IF DEF_CURRENT <> NIL THEN

                    IF (
                    (DOING)
                    AND
                    (DEF_CURRENT^.OBJECT_TYPE = PRI)
                    AND
                        (
                                (DEF_CURRENT^.PRIMARY <> WHICHPRIMARY)
                                OR
                                (DEF_CURRENT^.PRINUM <> WHICHPRINUM)
                                )
                        ) THEN

                                DONE    := TRUE;

                UNTIL (DONE) OR (DEF_CURRENT = NIL);

        END;    { DELETE_PRIMARY_SECTION }
        { ++

        INIT_DEF -- Clear out the definition and setup for a new one.

        This routine makes room to put a brand new definition in the linked list.

        CALLING SEQUENCE:

        INIT_DEF;

        INPUT PARAMETERS:

        none

        IMPLICIT INPUTS:

        none

        OUTPUT PARAMETERS:

        none

        IMPLICIT OUTPUTS:

        DEF_CURRENT
        DEF_HEAD

        ROUTINES CALLED:

        none

        ROUTINE VALUE:

        none

        SIGNALS:

        none

        SIDE EFFECTS:

        none

        -- }
```

```
PROCEDURE INIT_DEF;

BEGIN

    { +
    Clear out the list starting at the beginning (if not already empty).
    - }
    DEF_CURRENT := DEF_HEAD;

    IF DEF_CURRENT <> NIL THEN

    BEGIN

        REPEAT

            DELETE_CURRENT;

        UNTIL DEF_HEAD = DEF_TAIL;

        IF DEF_CURRENT <> NIL THEN

            DELETE_CURRENT;

    END;

END;    { INIT_DEF }
```

```
{ ++

INSERT_IN_ORDER -- Put DEF_SCRATCH into the list in proper order.

This routine places the line_object pointed to by def_scratch in the definition
linked list in its proper place.

CALLING SEQUENCE:

INSERT_IN_ORDER (COLLISION_ACTION);

INPUT PARAMETERS:

COLLISION_ACTION

IMPLICIT INPUTS:

DEF_CURRENT
DEF_TAIL
DEF_HEAD
DEF_SCRATCH

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

none

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
        [ASYNCHRONOUS] PROCEDURE INSERT_IN_ORDER (
                                COLLISION_ACTION    : INTEGER
                                );

        VAR
            BACKUP_WORKED           : BOOLEAN;

        BEGIN

            { +
            1st, a little conditioning.
            - }
            IF (
            (DEF_SCRATCH^.OBJECT_TYPE = PRI)
            AND
            (DEF_SCRATCH^.PRIMARY <> TITLE)
            ) THEN

                DEF_SCRATCH^.STRING.DSC$W_LENGTH        := 0;

            DEF_SCRATCH^.FORE   := NIL;
            DEF_SCRATCH^.BACK   := NIL;

            { +
            Now, find the proper place. Start looking at the previous line_object.
            - }
            BACKUP_WORKED           := FALSE;

            IF DEF_CURRENT <> NIL THEN

                IF DEF_CURRENT^.BACK <> NIL THEN

                BEGIN

                    DECR_CURRENT;

                    WHILE NOT (
                    (CURRENT_GT_TEST(DEF_SCRATCH^,TRUE))
                    OR
                    (CURRENT_EQ_TEST(DEF_SCRATCH^,TRUE))
                    OR
                    (DEF_CURRENT^.FORE = NIL)
                    ) DO

                        INCR_CURRENT;

                    BACKUP_WORKED           := (
                                            (
                                                (CURRENT_LT_TEST(DEF_SCRATCH^,TRUE))
                                                AND
                                                (DEF_CURRENT^.FORE = NIL)
                                            )
                                            OR
                                                (CURRENT_EQ_TEST(DEF_SCRATCH^,TRUE))
                                            );

                END;    { IF DEF_CURRENT^.BACK <> NIL }
        IF NOT BACKUP_WORKED THEN

        BEGIN

            { +
            The quick look didn't work, now scan the entire list.
            - }
            DEF_CURRENT       := DEF_HEAD;

            WHILE NOT (
            (CURRENT_GT_TEST(DEF_SCRATCH^,TRUE))
            OR
            (CURRENT_EQ_TEST(DEF_SCRATCH^,TRUE))
            OR
            (DEF_CURRENT^.FORE = NIL)
            ) DO

                INCR_CURRENT;

        END;        { IF NOT BACKUP_WORKED }

        { +
        Now insert it according to how it was (found).
        - }
        IF CURRENT_GT_TEST(DEF_SCRATCH^,TRUE) THEN

            INSERT_BEFORE_CURRENT

        ELSE IF CURRENT_EQ_TEST(DEF_SCRATCH^,TRUE) THEN

        BEGIN

            IF COLLISION_ACTION = REPLACE_OBJ THEN

                INSERT_AT_CURRENT

            ELSE IF COLLISION_ACTION = AFTER_OBJ THEN

                INSERT_AFTER_CURRENT;

            { IF COLLISION_ACTION = IGNORE_OBJ THEN 'NULL-STATEMENT' }

        END

        ELSE IF DEF_CURRENT^.FORE = NIL THEN

        BEGIN

            DEF_TAIL          := DEF_CURRENT;
            INSERT_AFTER_CURRENT;

        END;

    END;    { INSERT_IN_ORDER }
```

```
{ ++

FIND_OBJECT -- Locate a line_object in the definition list.

This function returns with DEF_CURRENT pointing to the desired line
object - if it finds it, in which case it's function value is true.

CALLING SEQUENCE:

BOOLEAN_VAR := FIND_OBJECT (LINE_OBJECT_TYPE,PRIMARY,PRINUM,SECONDARY,SECNUM);

INPUT PARAMETERS:

OBJECT_TYPE
PRIMARY
PRINUM
SECONDARY
SECNUM

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

DEF_CURRENT

ROUTINES CALLED:

none

ROUTINE VALUE:

TRUE/FALSE DEPENDING ON FOUND STATUS

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
FUNCTION FIND_OBJECT (
                        OBJ_TYP         : LINE_OBJECT_TYPE;
                        PRIM            : PRIMARY_TYPE;
                        PRIMNUM         : INTEGER;
                        SECO            : SECONDARY_TYPE;
                        SECONUM         : INTEGER
                        ) : BOOLEAN;

VAR
    TEST        : LINE_OBJECT;
    FOUND_IT    : BOOLEAN;
    PAST_IT     : BOOLEAN;

BEGIN

    { +
    Stuff test object for comparison routine.
    - }
    TEST.OBJECT_TYPE    := OBJ_TYP;
    TEST.PRIMARY        := PRIM;
    TEST.PRINUM         := PRIMNUM;
    TEST.SECONDARY      := SECO;
    TEST.SECNUM         := SECONUM;

    { +
    Start looking at head of list.
    - }
    DEF_CURRENT         := DEF_HEAD;
    FOUND_IT            := FALSE;
    PAST_IT             := FALSE;

    IF DEF_CURRENT <> NIL THEN

    BEGIN

        REPEAT

            FOUND_IT    := CURRENT_EQ_TEST (TEST,TRUE);
            PAST_IT     := CURRENT_GT_TEST (TEST,TRUE);

            IF NOT FOUND_IT THEN

                INCR_CURRENT;

        UNTIL (FOUND_IT) OR (PAST_IT) OR (DEF_CURRENT = NIL);

    END;

    { +
    Function value indicates whether we found it or not.
    - }
    FIND_OBJECT                 := FOUND_IT;

END;    { FIND_OBJECT }
```

```
{ ++

POINT_AT_DEFINITION -- Setup the list pointers.

This routine makes the list pointers point at the Definition Linked List.

CALLING SEQUENCE:

POINT_AT_DEFINITION;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

DEF_HEAD
DEF_TAIL
DEF_ANL_HEAD
DEF_ANL_TAIL
POINTING_DIRECTION

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

the current list is the definition list

-- }
```

```
[GLOBAL] PROCEDURE POINT_AT_DEFINITION;

BEGIN

    IF NOT POINTING_AT_DEFINITION THEN

    BEGIN

        DEF_ANL_HEAD            := DEF_HEAD;
        DEF_ANL_TAIL            := DEF_TAIL;
        DEF_HEAD                := DEF_SAVE_HEAD;
        DEF_TAIL                := DEF_SAVE_TAIL;

        POINTING_AT_DEFINITION  := TRUE;

    END;

END;    { POINT_AT_DEFINITION }
```

```
{ ++

POINT_AT_ANALYSIS -- Setup the list pointers.

This routine makes the list pointers point at the Analysis Linked List.

CALLING SEQUENCE:

POINT_AT_ANALYSIS;

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

none

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

DEF_HEAD
DEF_TAIL
DEF_ANL_HEAD
DEF_ANL_TAIL
POINTING_DIRECTION

ROUTINES CALLED:

none

ROUTINE VALUE:

none

SIGNALS:

none

SIDE EFFECTS:

the current list is the analysis list

-- }
```

```
        PROCEDURE POINT_AT_ANALYSIS;

        BEGIN

            IF POINTING_AT_DEFINITION THEN

            BEGIN

                DEF_SAVE_HEAD           := DEF_HEAD;
                DEF_SAVE_TAIL           := DEF_TAIL;
                DEF_HEAD                := DEF_ANL_HEAD;
                DEF_TAIL                := DEF_ANL_TAIL;

                POINTING_AT_DEFINITION  := FALSE;

            END;

        END;    { POINT_AT_ANALYSIS }
```

```
{ ++

EDF$LINE_PARSED -- Action routine for FDL$PARSE routine.

This routine stores into the definition database the values from FDL$PARSE.

CALLING SEQUENCE:

Called from the FDL$PARSE routine.

INPUT PARAMETERS:

none

IMPLICIT INPUTS:

FDL_BLOCK

OUTPUT PARAMETERS:

none

IMPLICIT OUTPUTS:

DEF_CURRENT
DEF_TAIL
DEF_HEAD
DEF_SCRATCH

ROUTINES CALLED:

MAKE_SCRATCH
LIB$SCOPY_DXDX

ROUTINE VALUE:

Always 1 (for success).

SIGNALS:

none

SIDE EFFECTS:

none

-- }
```

```
[ASYNCHRONOUS,GLOBAL] FUNCTION EDF$LINE_PARSED : INTEGER;

BEGIN

    { +
    This routine always succeeds.
    - }
    EDF$LINE_PARSED      := 1;

    { +
    Create a new line object to be added to the list.
    - }
    MAKE_SCRATCH;

    { +
    Get the control longword.
    - }
    TEMP_FDL3$TYPE       := FDL_BLOCK^[FDL$L_CTRL]::FDL3$TYPE;

    { +
    Completely ignore a line if it's an IDENT, or if the Warning bit is set.
    - }
    IF ( NOT (
    ((TEMP_FDL3$TYPE.FDL$V_NEWPRI)
    AND
    (FDL_BLOCK^[FDL$L_PRIMARY]::PRIMARY_TYPE = IDENT))
    OR
    (TEMP_FDL3$TYPE.FDL$V_WARNING)
    )) THEN

    WITH DEF_SCRATCH^ DO

    BEGIN

        { +
        Set the type of this line_object.
        - }
        IF TEMP_FDL3$TYPE.FDL$V_NEWPRI THEN

            OBJECT_TYPE := PRI

        ELSE

            OBJECT_TYPE := SEC;

        { +
        Check for a full line comment, as it is a 3rd object type.
        - }
        IF TEMP_FDL3$TYPE.FDL$V_LINECMT THEN

            OBJECT_TYPE := COM;

        { +
        Fetch the primary and secondary values.
        - }
        PRIMARY          := FDL_BLOCK^[FDL$L_PRIMARY]::PRIMARY_TYPE;
        SECONDARY        := FDL_BLOCK^[FDL$L_SECONDARY]::SECONDARY_TYPE;
```

```
{ +
Store the comment string if a comment was detected.
- }
IF TEMP_FDL3$TYPE.FDL$V_COMMENT OR TEMP_FDL3$TYPE.FDL$V_LINECMT THEN

BEGIN

    TEMP_DESCRIPTOR      := NULL_STRING;
    TEMP_DESCRIPTOR      := CVT_QUAD_DESC (
                                      FDL_BLOCK^[FDL$Q_COMMENT],
                                      FDL_BLOCK^[FDL$Q_COMMENT+1]
                                      );
    LIB$SCOPY_DXDX (TEMP_DESCRIPTOR,COMMENT);

END;

{ +
Store the string if it's an attribute with a string value.
- }
IF (

((NOT TEMP_FDL3$TYPE.FDL$V_NEWPRI) AND (SEC_TYPE[SECONDARY].STR))

OR
        { for positioning by file_name }
((NOT TEMP_FDL3$TYPE.FDL$V_NEWPRI) AND (SECONDARY = POSITION$))

OR

((TEMP_FDL3$TYPE.FDL$V_NEWPRI) AND (PRIMARY = TITLE))

) THEN

BEGIN

    TEMP_DESCRIPTOR      := NULL_STRING;
    TEMP_DESCRIPTOR      := CVT_QUAD_DESC (
                                      FDL_BLOCK^[FDL$Q_STRING],
                                      FDL_BLOCK^[FDL$Q_STRING+1]
                                      );
    LIB$SCOPY_DXDX (TEMP_DESCRIPTOR,STRING);

END;

{ +
Now stuff the new line_object with the remaining data from FDL$PARSE.
- }
IF PRIMARY IN [ AREA, KEY, ANALYSIS_OF_AREA, ANALYSIS_OF_KEY ] THEN

    PRINUM      := FDL_BLOCK^[FDL$L_PRINUM]

ELSE

    PRINUM      := 0;

IF SECONDARY IN [ SEG_LENGTH, SEG_POSITION ] THEN
```

```
                    SECNUM      := FDL_BLOCK^[FDL$L_SECNUM]

                { +
                Until RMS supports different types per segment,
                make the Type secondary that last one in the Key primary.
                (SECNUM is higher sorting priority than SECONDARY)
                - }
                ELSE IF SECONDARY = SEG_TYPE THEN

                    SECNUM      := 7

                ELSE

                    SECNUM      := 0;

        { ++

        IF RMS EVER IMPLEMENTS DIFFERENT DATA TYPES FOR EACH KEY SEGMENT,
        USE THE FOLLOWING CODE.

                IF SECONDARY IN [ SEG_LENGTH, SEG_POSITION, SEG_TYPE ] THEN

                    SECNUM      := FDL_BLOCK^[FDL$L_SECNUM]

                ELSE

                    SECNUM      := 0;

        -- }
```

```
{ +
Qualifiers values are more complicated.
- }
IF (
(SECONDARY = MT_PROTECTION)
OR
(SECONDARY = NULL_VALUE)
) THEN

BEGIN

    { +
    These two come back in a funny place.
    - }
    QUALIFIER    := 0;
    NUMBER       := FDL_BLOCK^[FDL$L_QUALIFIER];

END

ELSE

BEGIN

    { +
    See which secondary we have.
    - }
    CASE SECONDARY OF

        ORGANIZATION :

            CASE FDL_BLOCK^[FDL$L_QUALIFIER] OF

                FAB$C_IDX :     QUALIFIER       := FDL$C_IDX;
                FAB$C_REL :     QUALIFIER       := FDL$C_REL;
                FAB$C_SEQ :     QUALIFIER       := FDL$C_SEQ;

            OTHERWISE

                { NULL-STATEMENT } ;

            END;        { CASE FDL_BLOCK^[FDL$L_QUALIFIER] }

        FORMAT :

            CASE FDL_BLOCK^[FDL$L_QUALIFIER] OF

                FAB$C_FIX :     QUALIFIER       := FDL$C_FIX;
                FAB$C_STM :     QUALIFIER       := FDL$C_STM;
                FAB$C_STMCR :   QUALIFIER       := FDL$C_STMCR;
                FAB$C_STMLF :   QUALIFIER       := FDL$C_STMLF;
                FAB$C_UDF :     QUALIFIER       := FDL$C_UDF;
                FAB$C_VAR :     QUALIFIER       := FDL$C_VAR;
                FAB$C_VFC :     QUALIFIER       := FDL$C_VFC;

            OTHERWISE

                { NULL-STATEMENT } ;
```

```
                    END;            { CASE FDL_BLOCK^[FDL$L_QUALIFIER] }

                SEG_TYPE :

                    CASE FDL_BLOCK^[FDL$L_QUALIFIER] OF

                        XAB$C_BN2 :     QUALIFIER       := FDL$C_BN2;
                        XAB$C_BN4 :     QUALIFIER       := FDL$C_BN4;
                        XAB$C_BN8 :     QUALIFIER       := FDL$C_BN8;
                        XAB$C_PAC :     QUALIFIER       := FDL$C_PAC;
                        XAB$C_IN2 :     QUALIFIER       := FDL$C_IN2;
                        XAB$C_IN4 :     QUALIFIER       := FDL$C_IN4;
                        XAB$C_IN8 :     QUALIFIER       := FDL$C_IN8;
                        XAB$C_STG :     QUALIFIER       := FDL$C_STG;

                    OTHERWISE

                        { NULL-STATEMENT } ;

                    END;            { CASE FDL_BLOCK^[FDL$L_QUALIFIER] }

            OTHERWISE

                QUALIFIER       := FDL_BLOCK^[FDL$L_QUALIFIER];

            END;        { CASE SECONDARY }

            NUMBER          := FDL_BLOCK^[FDL$L_NUMBER];

        END;

        { +
        Now store the other information coming back from FDL$PARSE.
        - }
        IF ODD (FDL_BLOCK^[FDL$L_SWITCH]) THEN

            SWITCH          := TRUE

        ELSE

            SWITCH          := FALSE;

        OWNER_UIC           := FDL_BLOCK^[FDL$L_OWNER_UIC];
        PROT_MASK           := FDL_BLOCK^[FDL$L_PROTECTION]::CTRL_ARRAY;
        FID1                := FDL_BLOCK^[FDL$L_FID1];
        FID2                := FDL_BLOCK^[FDL$L_FID2];
        FID3                := FDL_BLOCK^[FDL$L_FID3];

        { +
        Now put def scratch into the linked list. Depending upon whether
        we're inputting an FDL file or a analysis file.
        CLUSTER_SIZE must go into both the analysis definition and the
        main definition.
        - }
        IF (
        (ANALYSIS_ONLY AND (PRIMARY IN [ ANALYSIS_OF_KEY, ANALYSIS_OF_AREA ]))
```

```
                OR
                ((NOT ANALYSIS_ONLY) AND (NOT (PRIMARY IN [ ANALYSIS_OF_KEY,
                ANALYSIS_OF_AREA ])))
                OR
                (SECONDARY IN [ CLUSTER_SIZE ])
                ) THEN

                    IF OBJECT_TYPE = COM THEN

                    BEGIN

                        DEF_CURRENT      := DEF_TAIL;
                        INSERT_AFTER_CURRENT;

                    END

                    ELSE

                        INSERT_IN_ORDER (IGNORE_OBJ);

            END;        { IF NOT IDENT PRIMARY; ALSO WITH DEF_SCRATCH^ DO }

        END;    { EDF$LINE_PARSED }

        END.
            { End of file: SRC$:EDFUTIL.PAS }
```

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

                                                            00000         .PSECT  $CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2

                                                5C  1B      00000 C.AAA:  .ASCII  <27><92>
                                                20  20      00002 C.AAB:  .ASCII  \  \
                        00  00  73  65  6E  69  6C  20      00004 C.AAC:  .ASCII  \ Lines\<0><0>
        00  5C  1B  3B  29  45  28  53  3B  70  50  1B      0000C C.AAD:  .ASCII  <27>\Pp;S(E);\<27><92><0>
56  3B  5D  30  32  33  2C  37  32  5B  50  70  50  1B      00018 C.AAE:  .ASCII  <27>\PpP[27,320];V(W(I0,S1,E,S[,479]))[+7\-
2C  5B  53  2C  45  2C  31  53  2C  30  49  28  57  28      00026                 \67];\<27><92><0>
1B  3B  5D  37  36  37  2B  5B  29  29  5D  39  37  34      00034
                                                00  5C      00042
        00  5C  1B  3B  29  45  28  53  3B  70  50  1B      00044 C.AAF:  .ASCII  <27>\Pp;S(E);\<27><92><0>
50  20  6F  6E  20  73  69  20  65  72  65  68  54  20      00050 C.AAG:  .ASCII  \ There is no Primary Key in the Current \-
20  6E  69  20  79  65  4B  20  79  72  61  6D  69  72      0005E                 \Definition. \
65  44  20  74  6E  65  72  72  75  43  20  65  68  74      0006C
                20  2E  6E  6F  69  74  69  6E  69  66      0007A
                                                08  38      00084 C.AAH:  .BYTE   ^X38,8
                                                            00086         .BLKB   2
                00000000   00000000   00000000  00000000    00088 C.AAI:  .LONG   0,0,0,0
                                            00  00  60      00098         .BYTE   ^X60,0,0
                                                            0009B         .BLKB   1
                00000000   00000800   00000000  00000000    0009C C.AAJ:  .LONG   0,0,^X800,0
                                            00  00  00      000AC         .BYTE   0,0,0
                                                            000AF         .BLKB   1
                                                00  18      000B0 C.AAK:  .BYTE   ^X18,0
                                                00  18      000B2 C.AAL:  .BYTE   ^X18,0

                                                       000C 00000 NUM_LEN:.WORD   ^M<R2,R3>                                ; 0166
                                    5C      04      BC   D0 00002         MOVL    @4(R12),NUMBER
                                                    00V  12 00006         BNEQ    2$                                       ; 0176
                                    50              01   D0 00008         MOVL    #1,NUM_LEN                               ; 0181
                                                    00V  11 0000B         BRB     10$
                                    51 3B9ACA00     8F   D0 0000D 2$:     MOVL    #1000000000,TEST_VAR                     ; 0190
                                    52              0A   D0 00014         MOVL    #10,TEST_LEN                             ; 0191
                                    53              5C   D0 00017         MOVL    NUMBER,R3                                ; 0193
                                                    00V  18 0001A         BGEQ    4$
                                    53              53   CE 0001C         MNEGL   R3,R3
                                    51              53   D1 0001F 4$:     CMPL    R3,TEST_VAR                              ; 0195
                                                    00V  18 00022         BGEQ    6$
                                                    52   D7 00024         DECL    TEST_LEN                                 ; 0197
                                    51              0A   C6 00026 6$:     DIVL2   #10,TEST_VAR                             ; 0199
                                    51              53   D1 00029         CMPL    R3,TEST_VAR
                                                    F1   19 0002C         BLSS    4$
                                                    5C   D5 0002E         TSTL    NUMBER                                   ; 0206
                                                    00V  18 00030         BGEQ    9$
                                                    52   D6 00032         INCL    TEST_LEN                                 ; 0208
                                    50              52   D0 00034 9$:     MOVL    TEST_LEN,NUM_LEN                         ; 0213
                                                         04 00037 10$:    RET                                              ; 0217

; Routine Size: 56 bytes,    Routine Base: $CODE + 00B4

                                                            00000 MAX_FACTOR:                                              ; 0267
                                                       001C 00000         .WORD   ^M<R2,R3,R4>
                                    50      04      BC   D0 00002         MOVL    @4(R12),BASE
                                    51      08      BC   D0 00006         MOVL    @8(R12),VALUE
```

```
                           5C          0C   BC   D0 0000A          MOVL    @12(R12),MAX
                           50               51   D1 0000E          CMPL    VALUE,BASE                        ; 0278
                                           00V   19 00011          BLSS    2$
                                            50   D5 00013          TSTL    BASE
                                           00V   12 00015          BNEQ    3$
                           52               50   D0 00017 2$:      MOVL    BASE,TEMP                         ; 0280
                                           00V   11 0001A          BRB     7$
                   52      51               50   C7 0001C 3$:      DIVL3   BASE,VALUE,TEMP                   ; 0286
         53        51      00               00   7A 00020          EMUL    #0,#0,VALUE,R3                    ; 0288
         53        53      53               50   7B 00025          EDIV    BASE,R3,R3,R3
                                            53   D5 0002A          TSTL    R3
                                           00V   18 0002C          BGEQ    4$
                           53               50   C0 0002E          ADDL2   BASE,R3
                                            53   D5 00031 4$:      TSTL    R3
                                           00V   13 00033          BEQL    6$
                                            52   D6 00035          INCL    TEMP                              ; 0290
                           52               50   C4 00037 6$:      MULL2   BASE,TEMP                         ; 0292
                           5C               52   D1 0003A 7$:      CMPL    TEMP,MAX                          ; 0296
                                           00V   15 0003D          BLEQ    9$
                           52               5C   D0 0003F          MOVL    MAX,TEMP                          ; 0298
                           50               52   D0 00042 9$:      MOVL    MAX_FACTOR,R0                     ; 0302
                                                 04 00045          RET

; Routine Size: 70 bytes,    Routine Base: $CODE + 000EC

                                                    00000 CALC_REC_OVERHEAD:                                 ; 0348
                                            0000 00000          .WORD   ^M<>
                           5C          04   BC   D0 00002          MOVL    @4(R12),INDEX_LEVEL
                                            50   D4 00006          CLRL    RECORD_OVERHEAD                   ; 0357
                             00000084G      EF   D5 00008          TSTL    IDATA+132                         ; 0362
                                           00V   13 0000E          BEQL    3$
                                            5C   D5 00010          TSTL    INDEX_LEVEL
                                           00V   12 00012          BNEQ    3$
                           50               0B   C0 00014          ADDL2   #11,RECORD_OVERHEAD               ; 0364
                 00V0000000CG   EF          00   E1 00017 3$:      BBC     #0,BDATA+12,5$                    ; 0369
                                            5C   D5 0001F          TSTL    INDEX_LEVEL
                                           00V   13 00021          BEQL    7$
                 00V0000000EG   EF          00   E1 00023 5$:      BBC     #0,BDATA+14,8$
                                            5C   D5 0002B          TSTL    INDEX_LEVEL
                                           00V   13 0002D          BEQL    8$
                           50               02   C0 0002F 7$:      ADDL2   #2,RECORD_OVERHEAD                ; 0375
                                            5C   D5 00032 8$:      TSTL    INDEX_LEVEL                       ; 0380
                                           00V   13 00034          BEQL    10$
                           50               04   C0 00036          ADDL2   #4,RECORD_OVERHEAD                ; 0382
                             00000084G      EF   D5 00039 10$:     TSTL    IDATA+132                         ; 0387
                                           00V   12 0003F          BNEQ    21$
                                            5C   D5 00041          TSTL    INDEX_LEVEL
                                           00V   12 00043          BNEQ    21$
                 00V00000000G   EF          00   E1 00045          BBC     #0,VARIABLE_RECORDS,14$           ; 0391
                                50          0B   C0 0004D          ADDL2   #11,RECORD_OVERHEAD               ; 0393
                                           00V   11 00050          BRB     15$
                                50          09   C0 00052 14$:     ADDL2   #9,RECORD_OVERHEAD                ; 0397
                 00V0000000DG   EF          00   E1 00055 15$:     BBC     #0,BDATA+13,17$                   ; 0399
                                50          03   C0 0005D          ADDL2   #3,RECORD_OVERHEAD                ; 0401
                 00V0000000CG   EF          00   E0 00060 17$:     BBS     #0,BDATA+12,19$                   ; 0403
                 00V0000000DG   EF          00   E1 00068          BBC     #0,BDATA+13,21$
                           50 000000D8G     EF   C0 00070 19$:     ADDL2   IDATA+216,RECORD_OVERHEAD         ; 0405
```

```
                                          04 00077 21$:     RET                                          ; 0411

; Routine Size: 120 bytes,    Routine Base: $CODE + 00132

                                             00000 CALC_BUC_OVERHEAD:                                    ; 0457
                                        0000 00000          .WORD   ^M<>
                  5C       04    BC       D0 00002          MOVL    @4(R12),INDEX_LEVEL
                                 00V      12 00006          BNEQ    2$                                   ; 0463
                  5C             10       D0 00008          MOVL    #16,CALC_BUC_OVERHEAD                ; 0465
                                 00V      11 0000B          BRB     3$
                  5C             12       D0 0000D 2$:      MOVL    #18,CALC_BUC_OVERHEAD                ; 0469
                  50             5C       D0 00010 3$:      MOVL    CALC_BUC_OVERHEAD,R0                 ; 0471
                                          04 00013          RET

; Routine Size: 20 bytes,    Routine Base: $CODE + 001AA

                                        0000 00000          .ENTRY  EDF$RESET_SCROLL,^M<>                ; 0520
                  5E             08       C2 00002          SUBL2   #8,SP
      00V00000000G EF            00       E0 00005          BBS     #0,AUTO_TUNE,8$                      ; 0524
      00V00000000G EF            00       E1 0000D          BBC     #0,REGIS,4$                          ; 0531
                       00000000G EF       D4 00015          CLRL    CHFFLAGS                             ; 0535
                       00000000G EF       B4 0001B          CLRW    OUT_LINE                             ; 0536
                       FFFFFE1B  EF       9F 00021          PUSHAB  C.AAA
                                 02       DD 00027          PUSHL   #2
                       00000000G EF       9F 00029          PUSHAB  OUT_LINE
                       000000FF  8F       DD 0002F          PUSHL   #255
           00000000G EF          04       FB 00035          CALLS   #4,PAS$WRITEV_STRING
                       00000000G EF       9F 0003C          PUSHAB  CHFFLAGS                             ; 0537
                       00000000G EF       9F 00042          PUSHAB  ONE
              F8   AD  0B2500FF  8F       D0 00048          MOVL    #186974463,-8(FP)
              FC   AD  00000000G EF       9E 00050          MOVAB   OUT_LINE,-4(FP)
                              F8 AD       9F 00058          PUSHAB  -8(FP)
           00000000G EF          03       FB 0005B          CALLS   #3,LIB$PUT_LINE
      00V00000000G EF            00       E1 00062 4$:      BBC     #0,SCROLLING_SET,7$                  ; 0544
                       00000000G EF       9F 0006A          PUSHAB  LINES_PER_PAGE                       ; 0546
                       00000000G EF       9F 00070          PUSHAB  LINE_ONE
           00000000G EF          02       FB 00076          CALLS   #2,LIB$SET_SCROLL
                                             0007D 7$:
        03 00000000G EF          00       E0 0007D 8$:      BBS     #0,FILE_CREATED,.+3                  ; 0553
                                 0000V    31 00085          BRW     12$
                       00000000G EF       B5 00088          TSTW    RES_OUTPUT_FILENAME_DESC
                                 03       1A 0008E          BGTRU   .+3
                                 0000V    31 00090          BRW     12$
           00000000G EF          02       D0 00093          MOVL    #2,CHFFLAGS                          ; 0561
                       00000000G EF       B4 0009A          CLRW    OUT_LINE                             ; 0562
                       00000000G EF       9F 000A0          PUSHAB  CRLF
                                 02       DD 000A6          PUSHL   #2
                       00000000G EF       9F 000A8          PUSHAB  OUT_LINE
                       000000FF  8F       DD 000AE          PUSHL   #255
           00000000G EF          04       FB 000B4          CALLS   #4,PAS$WRITEV_STRING
                  7E   00000000G EF       3C 000BB          MOVZWL  RES_OUTPUT_FILENAME_DESC,-(SP)
                                 00       DD 000C2          PUSHL   #0
                  50   00000004G EF       D0 000C4          MOVL    RES_OUTPUT_FILENAME_DESC+4,R0
                                 60       9F 000CB          PUSHAB  (R0)
                       000000FF  8F       DD 000CD          PUSHL   #255
                       00000000G EF       9F 000D3          PUSHAB  OUT_LINE
                       000000FF  8F       DD 000D9          PUSHL   #255
```

```
                                                            J  4
EDFUTIL                                           16-Sep-1984 00:51:37    VAX-11 Pascal V2.4-277                    Page  80
V04-000                 Generated Code             5-Sep-1984 13:38:55    DISK$VMSMASTER:[EDF.SRC]EDFUTIL.PAS;1 (59)

               00000000G  EF             06  FB 000DF          CALLS    #6,PAS$WRITEV_STRING
                             FFFFFD58    EF  9F 000E6          PUSHAB   C.AAB
                                         02  DD 000EC          PUSHL    #2
                             00000000G   EF  9F 000EE          PUSHAB   OUT_LINE
                             000000FF    8F  DD 000F4          PUSHL    #255
               00000000G  EF             04  FB 000FA          CALLS    #4,PAS$WRITEV_STRING
                             00000000G   EF  9F 00101          PUSHAB   LINES_SHOWN
                    00B4  CF             01  FB 00107          CALLS    #1,NUM_LEN
                                         50  DD 0010C          PUSHL    R0
                             00000000G   EF  DD 0010E          PUSHL    LINES_SHOWN
                             00000000G   EF  9F 00114          PUSHAB   OUT_LINE
                             000000FF    8F  DD 0011A          PUSHL    #255
               00000000G  EF             04  FB 00120          CALLS    #4,PAS$WRITEV_INTEGER
                             FFFFFD19    EF  9F 00127          PUSHAB   C.AAC
                                         06  DD 0012D          PUSHL    #6
                             00000000G   EF  9F 0012F          PUSHAB   OUT_LINE
                             000000FF    8F  DD 00135          PUSHL    #255
               00000000G  EF             04  FB 0013B          CALLS    #4,PAS$WRITEV_STRING
                             00000000G   EF  9F 00142          PUSHAB   CHFFLAGS                           ; 0565
                             00000000G   EF  9F 00148          PUSHAB   ONE
                      F8  AD 0B2500FF    8F  D0 0014E          MOVL     #186974463,-8(FP)
                      FC  AD 00000000G   EF  9E 00156          MOVAB    OUT_LINE,-4(FP)
                                     F8  AD  9F 0015E          PUSHAB   -8(FP)
               00000000G  EF             03  FB 00161          CALLS    #3,LIB$PUT_LINE
                                             04 00168 12$:     RET                                         ; 0569

; Routine Size: 361 bytes,    Routine Base: $CODE + 001BE

                                       0000 00000 CLEAR:       .WORD    ^M<>                               ; 0617
                          5C        04   BC  D0 00002          MOVL     @4(R12),DESTINATION
            03 00000000G  EF             00  E0 00006          BBS      #0,VIDEO_TERMINAL,.+3              ; 0626
                                 0000V   31 0000E              BRW      28$
            03 00000000G  EF             00  E1 00011          BBC      #0,AUTO_TUNE,.+3
                                 0000V   31 00019              BRW      28$
            03            00             5C  CF 0001C          CASEL    DESTINATION,#0,#3                  ; 0634
                                 0000V      00020              .DISPL   14$
                                 0000V      00022              .DISPL   8$
                                 0000V      00024              .DISPL   24$
                                 0000V      00026              .DISPL   3$
                                 0000V   31 00028              BRW      26$
            00V00000000G  EF             00  E1 0002B 3$:      BBC      #0,REGIS,5$                        ; 0646
                             FFFFFCAC    EF  9F 00033          PUSHAB   C.AAD                              ; 0648
                                         0B  DD 00039          PUSHL    #11
                             00000000G   EF  9F 0003B          PUSHAB   PAS$FV_OUTPUT
               00000000G  EF             03  FB 00041          CALLS    #3,PAS$WRITE_STRING
                             00000000G   EF  9F 00048          PUSHAB   PAS$FV_OUTPUT
               00000000G  EF             01  FB 0004E          CALLS    #1,PAS$WRITELN2
                             00000000G   EF  9F 00055 5$:      PUSHAB   COL_ONE                            ; 0650
                             00000000G   EF  9F 0005B          PUSHAB   LINE_ONE
               00000000G  EF             02  FB 00061          CALLS    #2,LIB$ERASE_PAGE
                                         01  DD 00068          PUSHL    #1                                 ; 0651
                                         20  DD 0006A          PUSHL    #32
                             00000000G   EF  9F 0006C          PUSHAB   PAS$FV_OUTPUT
               00000000G  EF             03  FB 00072          CALLS    #3,PAS$WRITE_CHAR
                             00000000G   EF  9F 00079          PUSHAB   PAS$FV_OUTPUT
               00000000G  EF             01  FB 0007F          CALLS    #1,PAS$WRITELN2
                             00000000G   EF  9F 00086          PUSHAB   COL_ONE                            ; 0652
```

```
                        00000000G   EF    9F 0008C          PUSHAB  LINE_ONE
            00000000G   EF          02    FB 00092          CALLS   #2,LIB$SET_CURSOR
                                 0000V    31 00099          BRW     27$
          00V00000000G  EF          00    E1 0009C 8$:      BBC     #0,REGIS,11$                                     ; 0660
                        FFFFFC47    EF    9F 000A4          PUSHAB  C.AAE                                            ; 0664
                                    2B    DD 000AA          PUSHL   #43
                        00000000G   EF    9F 000AC          PUSHAB  PAS$FV_OUTPUT
            00000000G   EF          03    FB 000B2          CALLS   #3,PAS$WRITE_STRING
                        00000000G   EF    9F 000B9          PUSHAB  PAS$FV_OUTPUT
            00000000G   EF          01    FB 000BF          CALLS   #1,PAS$WRITELN2
                        00000000G   EF    9F 000C6          PUSHAB  COL_ONE                                          ; 0667
                        00000000G   EF    9F 000CC          PUSHAB  PROMPT_LINE
            00000000G   EF          02    FB 000D2          CALLS   #2,LIB$SET_CURSOR
                                 0000V    31 000D9          BRW     27$
                        00000000G   EF    9F 000DC 11$:     PUSHAB  COL_ONE                                          ; 0673
                        00000000G   EF    9F 000E2          PUSHAB  LOWER_LINE
            00000000G   EF          02    FB 000E8          CALLS   #2,LIB$ERASE_PAGE
                                 0000V    31 000EF          BRW     27$
          00V00000000G  EF          00    E0 000F2 14$:     BBS     #0,FULL_PROMPT,16$                               ; 0677
          03 00000000G  EF          00    E0 000FA          BBS     #0,TEMP_FULL_PROMPT,.+3
                                 0000V    31 00102          BRW     27$
          00V00000000G  EF          00    E1 00105 16$:     BBC     #0,TEMP_FULL_PROMPT,18$                          ; 0681
                        666640A6    8F    DF 0010D          PUSHAF  #^F1.3                                           ; 0683
            00000000G   EF          01    FB 00113          CALLS   #1,LIB$WAIT
          00V00000000G  EF          00    E1 0011A 18$:     BBC     #0,REGIS,20$                                     ; 0692
                        FFFFFBF5    EF    9F 00122          PUSHAB  C.AAF                                            ; 0694
                                    0B    DD 00128          PUSHL   #11
                        00000000G   EF    9F 0012A          PUSHAB  PAS$FV_OUTPUT
            00000000G   EF          03    FB 00130          CALLS   #3,PAS$WRITE_STRING
                        00000000G   EF    9F 00137          PUSHAB  PAS$FV_OUTPUT
            00000000G   EF          01    FB 0013D          CALLS   #1,PAS$WRITELN2
                        00000000G   EF    9F 00144 20$:     PUSHAB  COL_ONE                                          ; 0696
                        00000000G   EF    9F 0014A          PUSHAB  LINE_ONE
            00000000G   EF          02    FB 00150          CALLS   #2,LIB$ERASE_PAGE
                                    01    DD 00157          PUSHL   #1                                               ; 0697
                                    20    DD 00159          PUSHL   #32
                        00000000G   EF    9F 0015B          PUSHAB  PAS$FV_OUTPUT
            00000000G   EF          03    FB 00161          CALLS   #3,PAS$WRITE_CHAR
                        00000000G   EF    9F 00168          PUSHAB  PAS$FV_OUTPUT
            00000000G   EF          01    FB 0016E          CALLS   #1,PAS$WRITELN2
                        00000000G   EF    9F 00175          PUSHAB  COL_ONE                                          ; 0698
                        00000000G   EF    9F 0017B          PUSHAB  LINE_ONE
            00000000G   EF          02    FB 00181          CALLS   #2,LIB$SET_CURSOR
                                   00V    11 00188          BRB     27$
                        0000001F    8F    DF 0018A 24$:     PUSHAL  #31                                              ; 0702
            00000000G   EF          01    FB 00190          CALLS   #1,QUERY
                                   00V    11 00197          BRB     27$
                                             00199 26$:
                                             00199 27$:
                                          04 00199 28$:     RET                                                      ; 0712

; Routine Size: 410 bytes,    Routine Base: $CODE + 00327

                                             00000 CVT_QUAD_DESC:                                                    ; 0759
                                        0000 00000          .WORD   ^M<>
                 5E                 08    C2 00002          SUBL2   #8,SP
                 50         04      BC    D0 00005          MOVL    @4(R12),LONG1
```

```
                    5C        08   BC   D0 00009             MOVL    @8(R12),LONG2
                      00000000G    EF   94 0000D             CLRB    QUAD_DESC                        ; 0770
           00000001G  EF           50   D0 00013             MOVL    LONG1,QUAD_DESC+1                ; 0771
           00000005G  EF           5C   D0 0001A             MOVL    LONG2,QUAD_DESC+5                ; 0772
           00000000G  EF           01   90 00021             MOVB    #1,QUAD_DESC                     ; 0777
                  F8  AD 00000001G EF   7D 00028             MOVQ    QUAD_DESC+1,CVT_QUAD_DESC        ; 0778
                  50           F8  AD   7D 00030             MOVQ    CVT_QUAD_DESC,R0                 ; 0782
                                        04 00034             RET

; Routine Size: 53 bytes,    Routine Base: $CODE + 004C1

                                           00000 SCAN_DEFINITION:                                     ; 0833
                                      0000 00000             .WORD   ^M<>
                    5C        04   BC   90 00002             MOVB    @4(R12),FATAL
           00000000G  EF 00000000G EF   D0 00006             MOVL    DEF_HEAD,DEF_CURRENT             ; 0840
                      00000000G    EF   94 00011             CLRB    FOUND_O                          ; 0841
                      00000000G    EF   94 00017             CLRB    FOUND_AREA                       ; 0842
                      00000000G    EF   94 0001D             CLRB    FOUND_KEY                        ; 0843
                      00000000G    EF   D4 00023             CLRL    LOW_AREA                         ; 0844
                      00000000G    EF   D4 00029             CLRL    HIGH_AREA                        ; 0845
                      00000000G    EF   D4 0002F             CLRL    LOW_KEY                          ; 0846
                      00000000G    EF   D4 00035             CLRL    HIGH_KEY                         ; 0847
                    50 00000000G   EF   D0 0003B 1$:         MOVL    DEF_CURRENT,R0                   ; 0851
                                   60   95 00042             TSTB    (R0)                             ; 0855
                                  00V   12 00044             BNEQ    11$
                    0B        19   A0   91 00046             CMPB    25(R0),#11
                                  00V   12 0004A             BNEQ    11$
                              1A   A0   D5 0004C             TSTL    26(R0)                           ; 0863
                                  00V   12 0004F             BNEQ    6$
           00000000G  EF           01   90 00051             MOVB    #1,FOUND_O                       ; 0865
           00000000G  EF           01   90 00058 6$:         MOVB    #1,FOUND_KEY                     ; 0867
           00000000G  EF      1A   A0   D1 0005F             CMPL    26(R0),LOW_KEY                   ; 0869
                                  00V   18 00067             BGEQ    8$
           00000000G  EF      1A   A0   D0 00069             MOVL    26(R0),LOW_KEY                   ; 0871
           00000000G  EF      1A   A0   D1 00071 8$:         CMPL    26(R0),HIGH_KEY                  ; 0873
                                  00V   15 00079             BLEQ    11$
           00000000G  EF      1A   A0   D0 0007B             MOVL    26(R0),HIGH_KEY                  ; 0875
                                   60   95 00083 11$:        TSTB    (R0)                             ; 0879
                                  00V   12 00085             BNEQ    18$
                    05        19   A0   91 00087             CMPB    25(R0),#5
                                  00V   12 0008B             BNEQ    18$
           00000000G  EF           01   90 0008D             MOVB    #1,FOUND_AREA                    ; 0887
           00000000G  EF      1A   A0   D1 00094             CMPL    26(R0),LOW_AREA                  ; 0889
                                  00V   18 0009C             BGEQ    15$
           00000000G  EF      1A   A0   D0 0009E             MOVL    26(R0),LOW_AREA                  ; 0891
           00000000G  EF      1A   A0   D1 000A6 15$:        CMPL    26(R0),HIGH_AREA                 ; 0893
                                  00V   15 000AE             BLEQ    18$
           00000000G  EF      1A   A0   D0 000B0             MOVL    26(R0),HIGH_AREA                 ; 0895
                    50 00000000G   EF   D0 000B8 18$:        MOVL    DEF_CURRENT,R0                   ; 0901
           00000000G  EF      01   A0   D0 000BF             MOVL    1(R0),DEF_CURRENT
                                   03   13 000C7             BEQL    .+3
                                 FF6F   31 000C9             BRW     1$
                   00V             5C   E8 000CC             BLBS    FATAL,21$                        ; 0905
                      00000000G    EF   D5 000CF             TSTL    HIGH_KEY
                                   03   12 000D5             BNEQ    .+3
                                0000V   31 000D7             BRW     26$
        03 00000000G  EF           00   E1 000DA 21$:        BBC     #0,FOUND_O,.+3
```

```
                              0000V   31 000E2           BRW      26$
                   00000000G  EF   9F 000E5           PUSHAB   SHIFT                                    ; 0913
                              04   DD 000EB           PUSHL    #4
                   00000000G  EF   9F 000ED           PUSHAB   PAS$FV_OUTPUT
       00000000G  EF          03   FB 000F3           CALLS    #3,PAS$WRITE_STRING
                   00000000G  EF   9F 000FA           PUSHAB   ANSI_REVERSE
                              04   DD 00100           PUSHL    #4
                   00000000G  EF   9F 00102           PUSHAB   PAS$FV_OUTPUT
       00000000G  EF          03   FB 00108           CALLS    #3,PAS$WRITE_STRING
                   FFFFFA45   EF   9F 0010F           PUSHAB   C.AAG
                              34   DD 00115           PUSHL    #52
                   00000000G  EF   9F 00117           PUSHAB   PAS$FV_OUTPUT
       00000000G  EF          03   FB 0011D           CALLS    #3,PAS$WRITE_STRING
                   00000000G  EF   9F 00124           PUSHAB   ANSI_RESET
                              04   DD 0012A           PUSHL    #4
                   00000000G  EF   9F 0012C           PUSHAB   PAS$FV_OUTPUT
       00000000G  EF          03   FB 00132           CALLS    #3,PAS$WRITE_STRING
                   00000000G  EF   9F 00139           PUSHAB   PAS$FV_OUTPUT
       00000000G  EF          01   FB 0013F           CALLS    #1,PAS$WRITELN2
     00V00000000G  EF         00   E1 00146           BBC      #0,AUTO_TUNE,24$                          ; 0917
                              00   DD 0014E           PUSHL    #0                                       ; 0919
                              00   DD 00150           PUSHL    #0
                              00   DD 00152           PUSHL    #0
                   00B3801C   8F   DD 00154           PUSHL    #11763740
       00000000G  EF          04   FB 0015A           CALLS    #4,LIB$STOP
                              00V  11 00161           BRB      26$
                   00004140   8F   DF 00163 24$:      PUSHAF   #^F3.0                                   ; 0925
       00000000G  EF          01   FB 00169           CALLS    #1,LIB$WAIT
                              00   DD 00170           PUSHL    #0                                       ; 0926
                              00   DD 00172           PUSHL    #0
                              00   DD 00174           PUSHL    #0
                   00B3804B   8F   DD 00176           PUSHL    #11763787
       00000000G  EF          04   FB 0017C           CALLS    #4,LIB$SIGNAL
                                   04 00183 26$:      RET                                               ; 0932

; Routine Size: 388 bytes,    Routine Base: $CODE + 004F6

                                      00000 PARSE_INPUT:                                                ; 0979
                              001C 00000              .WORD    ^M<R2,R3,R4>
              5E          08   C2 00002               SUBL2    #8,SP
              52     04   BC   D0 00005               MOVL     @4(R12),KEY_TABLE
              53     08   BC   D0 00009               MOVL     @8(R12),STATE_TABLE
              54     0C   BC   90 0000D               MOVB     @12(R12),DEFAULT_OK
              5C     10   BC   D0 00011               MOVL     @16(R12),DEFAULT_VALUE
       00000000G  EF 00000000G  EF   7D 00015         MOVQ     NULL_STRING,INPUT_DESC                   ; 0991
     00V00000000G  EF         00   E1 00020           BBC      #0,TAKE_DEFAULTS,4$                      ; 0996
                   00000104G  EF   D5 00028           TSTL     IDATA+260
                              00V  12 0002E           BNEQ     4$
              00V         54   E9 00030               BLBC     DEFAULT_OK,4$
              1F 00000000G  EF   D1 00033             CMPL     QTAB_OFFSET,#31
                              00V  12 0003A           BNEQ     5$
     00V00000000G  EF         00   E1 0003C 4$:       BBC      #0,AUTO_TUNE,8$
     03 00000000G  EF         00   E1 00044 5$:       BBC      #0,AUTO_TUNE,.+3                         ; 1012
                              0000V   31 0004C        BRW      12$
                   00000000G  EF   9F 0004F           PUSHAB   CRLF                                     ; 1016
                              02   DD 00055           PUSHL    #2
                   00000000G  EF   9F 00057           PUSHAB   PAS$FV_OUTPUT
```

```
         00000000G  EF           03   FB 0005D        CALLS    #3,PAS$WRITE_STRING
                    00000000G    EF   9F 00064        PUSHAB   PAS$FV_OUTPUT
         00000000G  EF           01   FB 0006A        CALLS    #1,PAS$WRITELN2
                    33334033     8F   DF 00071        PUSHAF   #^F0.7                              ; 1017
         00000000G  EF           01   FB 00077        CALLS    #1,LIB$WAIT
                               0000V  31 0007E        BRW      12$
       00V00000000G EF           30   E0 00081  8$:   BBS      #48,PAS$FV_INPUT,9$                 ; 1027
                    00000000G    EF   9F 00089        PUSHAB   PAS$FV_INPUT
         00000000G  EF           01   FB 0008F        CALLS    #1,PAS$LOOK_AHEAD
       00V00000000G EF           31   E0 00096  9$:   BBS      #49,PAS$FV_INPUT,11$
                    00000000G    EF   9F 0009E        PUSHAB   PAS$FV_INPUT                        ; 1031
         00000000G  EF           01   FB 000A4        CALLS    #1,PAS$RESET2
                                 00   DD 000AB        PUSHL    #0                                  ; 1032
                                 00   DD 000AD        PUSHL    #0
                                 00   DD 000AF        PUSHL    #0
                    00B3804B     8F   DD 000B1        PUSHL    #11763787
         00000000G  EF           04   FB 000B7        CALLS    #4,LIB$SIGNAL
                    000000FF     8F   DD 000BE  11$:  PUSHL    #255                                ; 1036
                    00000000G    EF   9F 000C4        PUSHAB   PAS$FV_INPUT
                    00000000G    EF   9F 000CA        PUSHAB   INPUT_STRING
         00000000G  EF           03   FB 000D0        CALLS    #3,PAS$READ_STRING
                    00000000G    EF   9F 000D7        PUSHAB   PAS$FV_INPUT
         00000000G  EF           01   FB 000DD        CALLS    #1,PAS$READLN2
                    00000000G    EF   9F 000E4        PUSHAB   CRLF                                ; 1037
                                 02   DD 000EA        PUSHL    #2
                    00000000G    EF   9F 000EC        PUSHAB   PAS$FV_OUTPUT
         00000000G  EF           03   FB 000F2        CALLS    #3,PAS$WRITE_STRING
                    00000000G    EF   9F 000F9        PUSHAB   PAS$FV_OUTPUT
         00000000G  EF           01   FB 000FF        CALLS    #1,PAS$WRITELN2
                F8  AD 010E00FF  8F   D0 00106        MOVL     #17694975,-8(FP)                    ; 1038
                FC  AD 00000000G EF   9E 0010E        MOVAB    INPUT_STRING,-4(FP)
                           F8    AD   9F 00116        PUSHAB   -8(FP)
                    00000000G    EF   9F 00119        PUSHAB   INPUT_DESC
         00000000G  EF           02   FB 0011F        CALLS    #2,STR$TRIM
                    00000000G    EF   9F 00126        PUSHAB   INPUT_DESC                          ; 1039
                    00000000G    EF   9F 0012C        PUSHAB   INPUT_DESC
         00000000G  EF           02   FB 00132        CALLS    #2,STR$UPCASE
       00V00000000G EF           00   E1 00139  12$:  BBC      #0,JOURNAL_ENABLED,17$              ; 1047
                    00000000G    EF   B5 00141        TSTW     INPUT_DESC                          ; 1049
                                00V   1B 00147        BLEQU    15$
                7E  00000000G    EF   3C 00149        MOVZWL   INPUT_DESC,-(SP)                    ; 1051
                                 00   DD 00150        PUSHL    #0
                50  00000004G    EF   D0 00152        MOVL     INPUT_DESC+4,R0
                                 60   9F 00159        PUSHAB   (R0)
                    000000FF     8F   DD 0015B        PUSHL    #255
                    00000000G    EF   9F 00161        PUSHAB   JOURNAL_FILE
         00000000G  EF           05   FB 00167        CALLS    #5,PAS$WRITE_STRING
                    00000000G    EF   9F 0016E        PUSHAB   JOURNAL_FILE
         00000000G  EF           01   FB 00174        CALLS    #1,PAS$WRITELN2
                                00V   11 0017B        BRB      16$
                    00000000G    EF   9F 0017D  15$:  PUSHAB   JOURNAL_FILE                        ; 1058
         00000000G  EF           01   FB 00183        CALLS    #1,PAS$WRITELN2
                                         0018A  16$:
                    00000000G    EF   B5 0018A  17$:  TSTW     INPUT_DESC                          ; 1063
                                00V   12 00190        BNEQ     22$
                    00V          54   E9 00192        BLBC     DEFAULT_OK,20$                      ; 1065
         00000000G  EF           5C   D0 00195        MOVL     DEFAULT_VALUE,INPUT_VALUE           ; 1067
```

```
                                0000V  31 0019C          BRW     28$
                                   00  DD 0019F  20$:    PUSHL   #0                                           ; 1073
                                   00  DD 001A1          PUSHL   #0
                                   00  DD 001A3          PUSHL   #0
                       00B38040    8F  DD 001A5          PUSHL   #11763776
         00000000G  EF             04  FB 001AB          CALLS   #4,LIB$SIGNAL
                                  00V  11 001B2          BRB     28$
         0000000CG  EF 00000004G   EF  D0 001B4  22$:    MOVL    INPUT_DESC+4,PARAM_BLOCK+12                  ; 1084
         00000008G  EF 00000000G   EF  3C 001BF          MOVZWL  INPUT_DESC,PARAM_BLOCK+8                     ; 1085
                                   52  DD 001CA          PUSHL   KEY_TABLE                                    ; 1087
                                   53  DD 001CC          PUSHL   STATE_TABLE
                       00000000G   EF  9F 001CE          PUSHAB  PARAM_BLOCK
         00000000G  EF             03  FB 001D4          CALLS   #3,LIB$TPARSE
         00000000G  EF             50  D0 001DB          MOVL    R0,ISTATUS
         00000000G  EF 00000020G   EF  D0 001E2          MOVL    PARAM_BLOCK+32,INPUT_VALUE                   ; 1093
         00000000G  EF 0000001CG   EF  D0 001ED          MOVL    PARAM_BLOCK+28,INPUT_NUMBER                  ; 1094
                    00V00000000G   EF  E8 001F8          BLBS    ISTATUS,28$                                  ; 1099
       00V00000006G  EF            00  E1 001FF          BBC     #0,PARAM_BLOCK+6,25$                         ; 1103
                                   00  DD 00207          PUSHL   #0                                           ; 1105
                                   00  DD 00209          PUSHL   #0
                                   00  DD 0020B          PUSHL   #0
                       00B38028    8F  DD 0020D          PUSHL   #11763752
         00000000G  EF             04  FB 00213          CALLS   #4,LIB$SIGNAL
                                  00V  11 0021A          BRB     28$
                                   00  DD 0021C  25$:    PUSHL   #0                                           ; 1109
                                   00  DD 0021E          PUSHL   #0
                                   00  DD 00220          PUSHL   #0
                       00B38030    8F  DD 00222          PUSHL   #11763760
         00000000G  EF             04  FB 00228          CALLS   #4,LIB$SIGNAL
                                       04 0022F  28$:    RET                                                  ; 1115

; Routine Size: 560 bytes,    Routine Base: $CODE + 0067A

                                          00000 NUMBER_INPUT:                                                 ; 1161
                                  000C  00000            .WORD   ^M<R2,R3>
                    5E             08  C2 00002          SUBL2   #8,SP
                    52         08  BC  90 00005          MOVB    @8(R12),DEFAULT_OK
                    53         0C  BC  D0 00009          MOVL    @12(R12),DEFAULT_VALUE
         00000000G  EF 00000000G   EF  7D 0000D          MOVQ    NULL_STRING,INPUT_DESC                       ; 1172
       00V00000000G  EF            00  E1 00018          BBC     #0,TAKE_DEFAULTS,3$                          ; 1177
                       00000104G   EF  D5 00020          TSTL    IDATA+260
                                  00V  12 00026          BNEQ    3$
                   00V             52  E8 00028          BLBS    DEFAULT_OK,4$
       00V00000000G  EF            00  E1 0002B  3$:     BBC     #0,AUTO_TUNE,7$
       03 00000000G  EF            00  E1 00033  4$:     BBC     #0,AUTO_TUNE,.+3                             ; 1191
                                0000V  31 0003B          BRW     11$
                       00000000G   EF  9F 0003E          PUSHAB  CRLF                                         ; 1195
                                   02  DD 00044          PUSHL   #2
                       00000000G   EF  9F 00046          PUSHAB  PAS$FV_OUTPUT
         00000000G  EF             03  FB 0004C          CALLS   #3,PAS$WRITE_STRING
                       00000000G   EF  9F 00053          PUSHAB  PAS$FV_OUTPUT
         00000000G  EF             01  FB 00059          CALLS   #1,PAS$WRITELN2
                       33334033    8F  DF 00060          PUSHAF  #^F0.7                                       ; 1196
         00000000G  EF             01  FB 00066          CALLS   #1,LIB$WAIT
                                0000V  31 0006D          BRW     11$
       00V00000000G  EF            30  E0 00070  7$:     BBS     #48,PAS$FV_INPUT,8$                          ; 1206
                       00000000G   EF  9F 00078          PUSHAB  PAS$FV_INPUT
```

```
     00000000G  EF                01  FB 0007E          CALLS   #1,PAS$LOOK_AHEAD
   00V00000000G  EF                31  E0 00085 8$:      BBS     #49,PAS$FV_INPUT,10$
                     00000000G  EF  9F 0008D          PUSHAB  PAS$FV_INPUT                              ; 1210
     00000000G  EF                01  FB 00093          CALLS   #1,PAS$RESET2
                                  00  DD 0009A          PUSHL   #0                                        ; 1211
                                  00  DD 0009C          PUSHL   #0
                                  00  DD 0009E          PUSHL   #0
                     00B3804B  8F  DD 000A0          PUSHL   #11763787
     00000000G  EF                04  FB 000A6          CALLS   #4,LIB$SIGNAL
                     000000FF  8F  DD 000AD 10$:     PUSHL   #255                                      ; 1215
                     00000000G  EF  9F 000B3          PUSHAB  PAS$FV_INPUT
                     00000000G  EF  9F 000B9          PUSHAB  INPUT_STRING
     00000000G  EF                03  FB 000BF          CALLS   #3,PAS$READ_STRING
                     00000000G  EF  9F 000C6          PUSHAB  PAS$FV_INPUT
     00000000G  EF                01  FB 000CC          CALLS   #1,PAS$READLN2
                     00000000G  EF  9F 000D3          PUSHAB  CRLF                                      ; 1216
                                  02  DD 000D9          PUSHL   #2
                     00000000G  EF  9F 000DB          PUSHAB  PAS$FV_OUTPUT
     00000000G  EF                03  FB 000E1          CALLS   #3,PAS$WRITE_STRING
                     00000000G  EF  9F 000E8          PUSHAB  PAS$FV_OUTPUT
     00000000G  EF                01  FB 000EE          CALLS   #1,PAS$WRITELN2
            F8  AD 010E00FF  8F  D0 000F5          MOVL    #17694975,-8(FP)                          ; 1217
            FC  AD 00000000G  EF  9E 000FD          MOVAB   INPUT_STRING,-4(FP)
                           F8  AD  9F 00105          PUSHAB  -8(FP)
                     00000000G  EF  9F 00108          PUSHAB  INPUT_DESC
     00000000G  EF                02  FB 0010E          CALLS   #2,STR$TRIM
                     00000000G  EF  9F 00115          PUSHAB  INPUT_DESC                                ; 1218
                     00000000G  EF  9F 0011B          PUSHAB  INPUT_DESC
     00000000G  EF                02  FB 00121          CALLS   #2,STR$UPCASE
     00000014G  EF 00000004G  EF  D0 00128          MOVL    INPUT_DESC+4,PARAM_BLOCK+20               ; 1219
     00000010G  EF 00000000G  EF  3C 00133          MOVZWL  INPUT_DESC,PARAM_BLOCK+16                 ; 1220
   00V00000000G  EF                00  E1 0013E 11$:     BBC     #0,JOURNAL_ENABLED,16$                    ; 1228
                     00000000G  EF  B5 00146          TSTW    INPUT_DESC                                ; 1230
                                 00V  1B 0014C          BLEQU   14$
                  7E 00000000G  EF  3C 0014E          MOVZWL  INPUT_DESC,-(SP)                          ; 1232
                                  00  DD 00155          PUSHL   #0
                  50 00000004G  EF  D0 00157          MOVL    INPUT_DESC+4,R0
                                  60  9F 0015E          PUSHAB  (R0)
                     000000FF  8F  DD 00160          PUSHL   #255
                     00000000G  EF  9F 00166          PUSHAB  JOURNAL_FILE
     00000000G  EF                05  FB 0016C          CALLS   #5,PAS$WRITE_STRING
                     00000000G  EF  9F 00173          PUSHAB  JOURNAL_FILE
     00000000G  EF                01  FB 00179          CALLS   #1,PAS$WRITELN2
                                 00V  11 00180          BRB     16$
                     00000000G  EF  9F 00182 14$:     PUSHAB  JOURNAL_FILE                              ; 1239
     00000000G  EF                01  FB 00188          CALLS   #1,PAS$WRITELN2
                     00000000G  EF  B5 0018F 16$:     TSTW    INPUT_DESC                                ; 1244
                                 00V  12 00195          BNEQ    21$
                 00V              52  E9 00197          BLBC    DEFAULT_OK,19$                            ; 1246
            04  BC                53  D0 0019A          MOVL    DEFAULT_VALUE,@4(R12)                     ; 1248
                                 00V  11 0019E          BRB     20$
                                  00  DD 001A0 19$:     PUSHL   #0                                        ; 1254
                                  00  DD 001A2          PUSHL   #0
                                  00  DD 001A4          PUSHL   #0
                     00B38040  8F  DD 001A6          PUSHL   #11763776
     00000000G  EF                04  FB 001AC          CALLS   #4,LIB$SIGNAL
                                 00V  11 001B3 20$:     BRB     24$
                                 04  AC   DD 001B5 21$:     PUSHL   4(R12)                                        ; 1265
                          00000000G  EF   9F 001B8          PUSHAB  INPUT_DESC
              00000000G  EF          02   FB 001BE          CALLS   #2,OTS$CVT_TI_L
              00000000G  EF          50   D0 001C5          MOVL    R0,ISTATUS
                        00V00000000G EF   E8 001CC          BLBS    ISTATUS,24$                                   ; 1270
                                     00   DD 001D3          PUSHL   #0                                            ; 1274
                                     00   DD 001D5          PUSHL   #0
                                     00   DD 001D7          PUSHL   #0
                           00B38030  8F   DD 001D9          PUSHL   #11763760
              00000000G  EF          04   FB 001DF          CALLS   #4,LIB$SIGNAL
              00000000G  EF      04  BC   D0 001E6 24$:     MOVL    @4(R12),INPUT_VALUE                           ; 1280
              00000000G  EF      04  BC   D0 001EE          MOVL    @4(R12),INPUT_NUMBER                          ; 1281
                                          04 001F6          RET                                                   ; 1283

; Routine Size: 503 bytes,    Routine Base: $CODE + 008AA

                                             00000 MAKE_SCRATCH:                                                  ; 1329
                                     003C 00000             .WORD   ^M<R2,R3,R4,R5>
                        5E       C0  AE   9E 00002          MOVAB   -64(SP),SP
                          00000040  8F    DD 00006          PUSHL   #64                                           ; 1336
              00000000G  EF          01   FB 0000C          CALLS   #1,PAS$NEW2
              00000000G  EF          50   D0 00013          MOVL    R0,DEF_SCRATCH
                     C0  AD 00000000G     EF D0 0001A       MOVL    DEF_SCRATCH,-64(FP)                           ; 1341
          C0   BD 00000000G  EF    0040  8F   28 00022      MOVC3   #64,LINE_OBJECT_TEMPLATE,@-64(FP)
                                          04 0002D          RET                                                   ; 1343

; Routine Size: 46 bytes,    Routine Base: $CODE + 00AA1

                                             00000 CURRENT_GT_TEST:                                               ; 1390
                                     003C 00000             .WORD   ^M<R2,R3,R4,R5>
                        5E       C0  AE   9E 00002          MOVAB   -64(SP),SP
          C0   AD       04  BC     0040  8F   28 00006      MOVC3   #64,@4(R12),TEST
                        5C       08  BC   90 0000E          MOVB    @8(R12),EXACT_COMPARISON
                        50           94 00012               CLRB    CURRENT_GT_TEST                               ; 1397
               51 00000000G  EF      19   C1 00014          ADDL3   #25,DEF_CURRENT,R1                            ; 1402
                        51           61   9A 0001C          MOVZBL  (R1),R1
                        51 00000000GEF41  9E 0001F          MOVAB   PRI_SEQ[R1],R1
                        52       D9  AD   9A 00027          MOVZBL  TEST+25,R2
                        52 00000000GEF42  9E 0002B          MOVAB   PRI_SEQ[R2],R2
                        53 00000000G EF   D0 00033          MOVL    DEF_CURRENT,R3
                        53       1A  A3   D0 0003A          MOVL    26(R3),R3
                        00V          5C   E9 0003E          BLBC    EXACT_COMPARISON,16$
                        62           61   91 00041          CMPB    (R1),(R2)                                     ; 1406
                                     00V  1B 00044          BLEQU   3$
                        50           01   90 00046          MOVB    #1,CURRENT_GT_TEST                            ; 1410
                   DA   AD           53   D1 00049 3$:      CMPL    R3,TEST+26                                    ; 1412
                                     00V  15 0004D          BLEQ    6$
                        62           61   91 0004F          CMPB    (R1),(R2)
                                     00V  12 00052          BNEQ    6$
                        50           01   90 00054          MOVB    #1,CURRENT_GT_TEST                            ; 1418
                   DA   AD           53   D1 00057 6$:      CMPL    R3,TEST+26                                    ; 1420
                                     00V  12 0005B          BNEQ    10$
                        62           61   91 0005D          CMPB    (R1),(R2)
                                     00V  12 00060          BNEQ    10$
                        5C 00000000G EF   D0 00062          MOVL    DEF_CURRENT,R12
                   DF   AD       1F  AC   D1 00069          CMPL    31(R12),TEST+31
                                     00V  15 0006E          BLEQ    10$
                     50                  01   90 00070              MOVB    #1,CURRENT_GT_TEST                        ; 1428
                     5C 00000000G        EF   D0 00073 10$:         MOVL    DEF_CURRENT,R12                           ; 1430
               DE    AD           1E     AC   91 0007A              CMPB    30(R12),TEST+30
                                         00V  1B 0007F              BLEQU   22$
               DA    AD                  53   D1 00081              CMPL    R3,TEST+26
                                         00V  12 00085              BNEQ    22$
                     62                  61   91 00087              CMPB    (R1),(R2)
                                         00V  12 0008A              BNEQ    22$
                     5C 00000000G        EF   D0 0008C              MOVL    DEF_CURRENT,R12
               DF    AD           1F     AC   D1 00093              CMPL    31(R12),TEST+31
                                         00V  12 00098              BNEQ    22$
                     50                  01   90 0009A              MOVB    #1,CURRENT_GT_TEST                        ; 1440
                                         00V  11 0009D              BRB     22$
                     62                  61   91 0009F 16$:         CMPB    (R1),(R2)                                 ; 1448
                                         00V  1B 000A2              BLEQU   18$
                     50                  01   90 000A4              MOVB    #1,CURRENT_GT_TEST                        ; 1452
               DA    AD                  53   D1 000A7 18$:         CMPL    R3,TEST+26                                ; 1454
                                         00V  15 000AB              BLEQ    22$
                     62                  61   91 000AD              CMPB    (R1),(R2)
                                         00V  12 000B0              BNEQ    22$
                     50                  01   90 000B2              MOVB    #1,CURRENT_GT_TEST                        ; 1460
                                              04 000B5 22$:         RET                                               ; 1464

; Routine Size: 182 bytes,     Routine Base: $CODE + 00ACF

                                                 00000 CURRENT_LT_TEST:                                               ; 1511
                                            003C 00000              .WORD   ^M<R2,R3,R4,R5>
                     5E           C0     AE   9E 00002              MOVAB   -64(SP),SP
         C0    AD    04     BC  0040     8F   28 00006              MOVC3   #64,@4(R12),TEST
                     5C           08     BC   90 0000E              MOVB    @8(R12),EXACT_COMPARISON
                                         50   94 00012              CLRB    CURRENT_LT_TEST                           ; 1518
            51 00000000G    EF           19   C1 00014              ADDL3   #25,DEF_CURRENT,R1                        ; 1523
                            51           61   9A 0001C              MOVZBL  (R1),R1
                            51 00000000GEF41  9E 0001F              MOVAB   PRI_SEQ[R1],R1
                            52           D9   AD   9A 00027         MOVZBL  TEST+25,R2
                            52 00000000GEF42  9E 0002B              MOVAB   PRI_SEQ[R2],R2
                            53 00000000G     EF   D0 00033          MOVL    DEF_CURRENT,R3
                            53           1A   A3   D0 0003A         MOVL    26(R3),R3
                            00V               5C   E9 0003E         BLBC    EXACT_COMPARISON,16$
                            62                61   91 00041         CMPB    (R1),(R2)                                 ; 1527
                                         00V  1E 00044              BGEQU   3$
                     50                  01   90 00046              MOVB    #1,CURRENT_LT_TEST                        ; 1531
               DA    AD                  53   D1 00049 3$:          CMPL    R3,TEST+26                                ; 1533
                                         00V  18 0004D              BGEQ    6$
                     62                  61   91 0004F              CMPB    (R1),(R2)
                                         00V  12 00052              BNEQ    6$
                     50                  01   90 00054              MOVB    #1,CURRENT_LT_TEST                        ; 1539
               DA    AD                  53   D1 00057 6$:          CMPL    R3,TEST+26                                ; 1541
                                         00V  12 0005B              BNEQ    10$
                     62                  61   91 0005D              CMPB    (R1),(R2)
                                         00V  12 00060              BNEQ    10$
                     5C 00000000G        EF   D0 00062              MOVL    DEF_CURRENT,R12
               DF    AD           1F     AC   D1 00069              CMPL    31(R12),TEST+31
                                         00V  18 0006E              BGEQ    10$
                     50                  01   90 00070              MOVB    #1,CURRENT_LT_TEST                        ; 1549
                     5C 00000000G        EF   D0 00073 10$:         MOVL    DEF_CURRENT,R12                           ; 1551
               DE    AD           1E     AC   91 0007A              CMPB    30(R12),TEST+30
```

```
                                      00V   1E 0007F          BGEQU   22$
                    DA    AD          53    D1 00081          CMPL    R3,TEST+26
                                      00V   12 00085          BNEQ    22$
                          62          61    91 00087          CMPB    (R1),(R2)
                                      00V   12 0008A          BNEQ    22$
                          5C 00000000G EF   D0 0008C          MOVL    DEF_CURRENT,R12
                    DF    AD       1F AC    D1 00093          CMPL    31(R12),TEST+31
                                      00V   12 00098          BNEQ    22$
                          50          01    90 0009A          MOVB    #1,CURRENT_LT_TEST                      ; 1561
                                      00V   11 0009D          BRB     22$
                          62          61    91 0009F 16$:     CMPB    (R1),(R2)                               ; 1569
                                      00V   1E 000A2          BGEQU   18$
                          50          01    90 000A4          MOVB    #1,CURRENT_LT_TEST                      ; 1573
                    DA    AD          53    D1 000A7 18$:     CMPL    R3,TEST+26                              ; 1575
                                      00V   18 000AB          BGEQ    22$
                          62          61    91 000AD          CMPB    (R1),(R2)
                                      00V   12 000B0          BNEQ    22$
                          50          01    90 000B2          MOVB    #1,CURRENT_LT_TEST                      ; 1581
                                            04 000B5 22$:     RET                                             ; 1585

; Routine Size: 182 bytes,    Routine Base: $CODE + 00B85

                                               00000 CURRENT_EQ_TEST:                                         ; 1632
                                          003C 00000          .WORD   ^M<R2,R3,R4,R5>
                          5E       C0 AE    9E 00002          MOVAB   -64(SP),SP
              C0    AD    04 BC  0040 8F    28 00006          MOVC3   #64,@4(R12),TEST
                          5C       08 BC    90 0000E          MOVB    @8(R12),EXACT_COMPARISON
                    50 00000000G EF   19    C1 00012          ADDL3   #25,DEF_CURRENT,R0                      ; 1642
                                      51    94 0001A          CLRB    R1
                          60       D9 AD    91 0001C          CMPB    TEST+25,(R0)
                                      00V   12 00020          BNEQ    2$
                                      51    96 00022          INCB    R1
                    50 00000000G EF   1A    C1 00024 2$:      ADDL3   #26,DEF_CURRENT,R0
                                      52    94 0002C          CLRB    R2
                          60       DA AD    D1 0002E          CMPL    TEST+26,(R0)
                                      00V   12 00032          BNEQ    4$
                                      52    96 00034          INCB    R2
                          00V         5C    E9 00036 4$:      BLBC    EXACT_COMPARISON,13$
                          5C 00000000G EF   D0 00039          MOVL    DEF_CURRENT,R12                         ; 1644
                    1F    AC       DF AD    D1 00040          CMPL    TEST+31,31(R12)
                                      00V   12 00045          BNEQ    11$
                          5C 00000000G EF   D0 00047          MOVL    DEF_CURRENT,R12
                    1E    AC       DE AD    91 0004E          CMPB    TEST+30,30(R12)
                                      00V   12 00053          BNEQ    11$
                          00V         52    E9 00055          BLBC    R2,11$
                          5C 00000000G EF   D0 00058          MOVL    DEF_CURRENT,R12
                          6C       C0 AD    91 0005F          CMPB    TEST,(R12)
                                      00V   12 00063          BNEQ    11$
                          00V         51    E9 00065          BLBC    R1,11$
                          5C          01    90 00068          MOVB    #1,CURRENT_EQ_TEST
                                      00V   11 0006B          BRB     14$
                                      5C    94 0006D 11$:     CLRB    CURRENT_EQ_TEST
                                      00V   11 0006F          BRB     14$
                          52          52    92 00071 13$:     MCOMB   R2,R2                                   ; 1660
                    5C    51          52    8B 00074          BICB3   R2,R1,CURRENT_EQ_TEST
                          50          5C    90 00078 14$:     MOVB    CURRENT_EQ_TEST,R0                      ; 1668
                                            04 0007B          RET
```

```
; Routine Size: 124 bytes,    Routine Base: $CODE + 00C3B

                                          00000 INSERT_BEFORE_CURRENT:                                             ; 1716
                                     0000 00000              .WORD    ^M<>
      00000000G  EF 00000000G  EF     D1 00002              CMPL     DEF_CURRENT,DEF_HEAD                         ; 1723
                                 00V  12 0000D              BNEQ     2$
      00000000G  EF 00000000G  EF     D0 0000F              MOVL     DEF_SCRATCH,DEF_HEAD                         ; 1725
                 50 00000000G  EF     D0 0001A 2$:          MOVL     DEF_CURRENT,R0                               ; 1730
      00000000G  EF        05  A0     D0 00021              MOVL     5(R0),DEF_PRED
                 50 00000000G  EF     D0 00029              MOVL     DEF_SCRATCH,R0                               ; 1731
             01  A0 00000000G  EF     D0 00030              MOVL     DEF_CURRENT,1(R0)
                 50 00000000G  EF     D0 00038              MOVL     DEF_SCRATCH,R0                               ; 1732
             05  A0 00000000G  EF     D0 0003F              MOVL     DEF_PRED,5(R0)
                    00000000G  EF     D5 00047              TSTL     DEF_PRED                                     ; 1734
                                 00V  13 0004D              BEQL     4$
                 50 00000000G  EF     D0 0004F              MOVL     DEF_PRED,R0                                  ; 1736
             01  A0 00000000G  EF     D0 00056              MOVL     DEF_SCRATCH,1(R0)
                 50 00000000G  EF     D0 0005E 4$:          MOVL     DEF_CURRENT,R0                               ; 1738
             05  A0 00000000G  EF     D0 00065              MOVL     DEF_SCRATCH,5(R0)
      00000000G  EF 00000000G  EF     D0 0006D              MOVL     DEF_SCRATCH,DEF_CURRENT                      ; 1743
                                      04 00078              RET                                                   ; 1745

; Routine Size: 121 bytes,    Routine Base: $CODE + 00CB7

                                          00000 INSERT_AT_CURRENT:                                                 ; 1793
                                     0000 00000              .WORD    ^M<>
      00000000G  EF 00000000G  EF     D1 00002              CMPL     DEF_CURRENT,DEF_HEAD                         ; 1800
                                 00V  12 0000D              BNEQ     2$
      00000000G  EF 00000000G  EF     D0 0000F              MOVL     DEF_SCRATCH,DEF_HEAD                         ; 1802
      00000000G  EF 00000000G  EF     D1 0001A 2$:          CMPL     DEF_CURRENT,DEF_TAIL                         ; 1804
                                 00V  12 00025              BNEQ     4$
      00000000G  EF 00000000G  EF     D0 00027              MOVL     DEF_SCRATCH,DEF_TAIL                         ; 1806
                 50 00000000G  EF     D0 00032 4$:          MOVL     DEF_CURRENT,R0                               ; 1811
      00000000G  EF        05  A0     D0 00039              MOVL     5(R0),DEF_PRED
                 50 00000000G  EF     D0 00041              MOVL     DEF_CURRENT,R0                               ; 1812
      00000000G  EF        01  A0     D0 00048              MOVL     1(R0),DEF_SUCC
                 50 00000000G  EF     D0 00050              MOVL     DEF_SCRATCH,R0                               ; 1813
                 51 00000000G  EF     D0 00057              MOVL     DEF_CURRENT,R1
             01  A0        01  A1     D0 0005E              MOVL     1(R1),1(R0)
                 50 00000000G  EF     D0 00063              MOVL     DEF_SCRATCH,R0                               ; 1814
                 51 00000000G  EF     D0 0006A              MOVL     DEF_CURRENT,R1
             05  A0        05  A1     D0 00071              MOVL     5(R1),5(R0)
                    00000000G  EF     D5 00076              TSTL     DEF_PRED                                     ; 1816
                                 00V  13 0007C              BEQL     6$
                 50 00000000G  EF     D0 0007E              MOVL     DEF_PRED,R0                                  ; 1818
             01  A0 00000000G  EF     D0 00085              MOVL     DEF_SCRATCH,1(R0)
                    00000000G  EF     D5 0008D 6$:          TSTL     DEF_SUCC                                     ; 1820
                                 00V  13 00093              BEQL     8$
                 50 00000000G  EF     D0 00095              MOVL     DEF_SUCC,R0                                  ; 1822
             05  A0 00000000G  EF     D0 0009C              MOVL     DEF_SCRATCH,5(R0)
                    00000000G  EF     DD 000A4 8$:          PUSHL    DEF_CURRENT                                  ; 1827
      00000000G  EF               01  FB 000AA              CALLS    #1,PAS$DISPOSE2
      00000000G  EF 00000000G  EF     D0 000B1              MOVL     DEF_SCRATCH,DEF_CURRENT                      ; 1829
                                      04 000BC              RET                                                   ; 1831

; Routine Size: 189 bytes,    Routine Base: $CODE + 00D30
```

```
                                          00000 INSERT_AFTER_CURRENT:                                              ; 1879
                                     0000 00000              .WORD    ^M<>
       00000000G  EF 00000000G  EF    D1 00002              CMPL     DEF_CURRENT,DEF_TAIL                         ; 1886
                               00V    12 0000D              BNEQ     2$
       00000000G  EF 00000000G  EF    D0 0000F              MOVL     DEF_SCRATCH,DEF_TAIL                         ; 1888
                  50 00000000G  EF    D0 0001A 2$:          MOVL     DEF_CURRENT,R0                               ; 1893
       00000000G  EF         01  A0    D0 00021              MOVL     1(R0),DEF_SUCC
                  50 00000000G  EF    D0 00029              MOVL     DEF_SCRATCH,R0                               ; 1894
              01  A0 00000000G  EF    D0 00030              MOVL     DEF_SUCC,1(R0)
                  50 00000000G  EF    D0 00038              MOVL     DEF_SCRATCH,R0                               ; 1895
              05  A0 00000000G  EF    D0 0003F              MOVL     DEF_CURRENT,5(R0)
                  50 00000000G  EF    D0 00047              MOVL     DEF_CURRENT,R0                               ; 1896
              01  A0 00000000G  EF    D0 0004E              MOVL     DEF_SCRATCH,1(R0)
                     00000000G  EF    D5 00056              TSTL     DEF_SUCC                                     ; 1898
                               00V    13 0005C              BEQL     4$
                  50 00000000G  EF    D0 0005E              MOVL     DEF_SUCC,R0                                  ; 1900
              05  A0 00000000G  EF    D0 00065              MOVL     DEF_SCRATCH,5(R0)
       00000000G  EF 00000000G  EF    D0 0006D 4$:          MOVL     DEF_SCRATCH,DEF_CURRENT                      ; 1905
                                      04 00078              RET                                                   ; 1907

; Routine Size: 121 bytes,    Routine Base: $CODE + 00DED

                                          00000 INCR_CURRENT:                                                      ; 1953
                                     0000 00000              .WORD    ^M<>
                     00000000G  EF    D5 00002              TSTL     DEF_CURRENT                                  ; 1960
                               00V    13 00008              BEQL     2$
                  50 00000000G  EF    D0 0000A              MOVL     DEF_CURRENT,R0                               ; 1962
       00000000G  EF         01  A0    D0 00011              MOVL     1(R0),DEF_CURRENT
                                      04 00019 2$:          RET                                                   ; 1964

; Routine Size: 26 bytes,    Routine Base: $CODE + 00E66

                                          00000 DECR_CURRENT:                                                      ; 2010
                                     0000 00000              .WORD    ^M<>
                     00000000G  EF    D5 00002              TSTL     DEF_CURRENT                                  ; 2017
                               00V    13 00008              BEQL     2$
                  50 00000000G  EF    D0 0000A              MOVL     DEF_CURRENT,R0                               ; 2019
       00000000G  EF         05  A0    D0 00011              MOVL     5(R0),DEF_CURRENT
                                      04 00019 2$:          RET                                                   ; 2021

; Routine Size: 26 bytes,    Routine Base: $CODE + 00E80

                                          00000 NEW_IDENT_LINE:                                                    ; 2075
                                     000C 00000              .WORD    ^M<R2,R3>
                  5E              1C    C2 00002              SUBL2    #28,SP
            0AA1  CF              00    FB 00005              CALLS    #0,MAKE_SCRATCH                              ; 2086
              F8  AD 010E0014     8F    D0 0000A              MOVL     #17694740,-8(FP)                             ; 2091
              FC  AD         E4   AD    9E 00012              MOVAB    DATE_STRING,-4(FP)
                             F8   AD    9F 00017              PUSHAB   -8(FP)
       00000000G  EF              01    FB 0001A              CALLS    #1,LIB$DATE_TIME
                  50              01    D0 00021              MOVL     #1,R0                                        ; 2096
                  5C              50    D0 00024 2$:          MOVL     R0,I
       FFFFFFFFGEF4C          E3 AD4C    90 00027              MOVB     DATE_STRING-1[I],IDENT_STRING-1[I]           ; 2098
    EF            50              14    F3 00031              AOBLEQ   #20,R0,2$
                  52 00000000G  EF    D0 00035              MOVL     DEF_SCRATCH,R2                               ; 2103
       00000000G  EF 00000000G  EF    7D 0003C              MOVQ     NUL[_STRING,TEMP_DESCRIPTOR                  ; 2107
```

```
               000000FF      8F DD 00047       PUSHL   #255                                    ; 2108
00000000G  EF                01 FB 0004D       CALLS   #1,PAS$NEW2
00000004G  EF                50 D0 00054       MOVL    R0,TEMP_DESCRIPTOR+4
00000000G  EF 00000000G      EF B0 0005B       MOVW    IDENT_STRING_LENGTH,TEMP_DESCRIPTOR     ; 2110
                             62 94 00066       CLRB    (R2)                                    ; 2111
       19  A2                09 90 00068       MOVB    #9,25(R2)                               ; 2112
           50                01 D0 0006C       MOVL    #1,R0                                   ; 2114
           51 00000000G      EF D0 0006F       MOVL    IDENT_STRING_LENGTH,R1
           51                50 D1 00076       CMPL    R0,R1
                            00V 15 00079       BLEQ    5$
                            00V 11 0007B       BRB     6$
                             50 D6 0007D 4$:   INCL    R0
           5C                50 D0 0007F 5$:   MOVL    R0,I
           53 00000004G      EF D0 00082       MOVL    TEMP_DESCRIPTOR+4,R3                    ; 2116
          FF A34C FFFFFFFFGEF4C 90 00089       MOVB    IDENT_STRING-1[I],-1(R3)[I]
           51                50 D1 00093       CMPL    R0,R1
                             E5 19 00096       BLSS    4$
                     11      A2 9F 00098 6$:   PUSHAB  17(R2)                                  ; 2118
              00000000G      EF 9F 0009B       PUSHAB  TEMP_DESCRIPTOR
00000000G  EF                02 FB 000A1       CALLS   #2,LIB$SCOPY_DXDX
              00000004G      EF DD 000A8       PUSHL   TEMP_DESCRIPTOR+4                       ; 2119
00000000G  EF                01 FB 000AE       CALLS   #1,PAS$DISPOSE2
00000000G  EF 00000000G      EF D0 000B5       MOVL    DEF_SCRATCH,DEF_CURRENT                 ; 2126
00000000G  EF 00000000G      EF D0 000C0       MOVL    DEF_SCRATCH,DEF_HEAD                    ; 2127
00000000G  EF 00000000G      EF D0 000CB       MOVL    DEF_SCRATCH,DEF_TAIL                    ; 2128
                                04 000D6       RET                                             ; 2130

; Routine Size: 215 bytes,    Routine Base: $CODE + 00E9A

                                   00000 DELETE_CURRENT:                                       ; 2180
                              0000 00000       .WORD   ^M<>
           50 00000000G      EF D0 00002       MOVL    DEF_CURRENT,R0                          ; 2184
           0F        19      A0 91 00009       CMPB    25(R0),#15
                            00V 12 0000D       BNEQ    5$
           50 00000000G      EF D0 0000F       MOVL    DEF_CURRENT,R0                          ; 2191
                     01      A0 D5 00016       TSTL    1(R0)
                            00V 12 00019       BNEQ    3$
              00000000G      EF DD 0001B       PUSHL   DEF_CURRENT                             ; 2195
00000000G  EF                01 FB 00021       CALLS   #1,PAS$DISPOSE2
     0E9A  CF                00 FB 00028       CALLS   #0,NEW_IDENT_LINE                       ; 2196
                          0000V 31 0002D       BRW     27$
           50 00000000G      EF D0 00030 3$:   MOVL    DEF_CURRENT,R0                          ; 2204
00000000G  EF        01      A0 D0 00037       MOVL    1(R0),DEF_HEAD
           50 00000000G      EF D0 0003F       MOVL    DEF_HEAD,R0                             ; 2205
                     05      A0 D4 00046       CLRL    5(R0)
              00000000G      EF DD 00049       PUSHL   DEF_CURRENT                             ; 2206
00000000G  EF                01 FB 0004F       CALLS   #1,PAS$DISPOSE2
00000000G  EF 00000000G      EF D0 00056       MOVL    DEF_HEAD,DEF_CURRENT                    ; 2207
                          0000V 31 00061       BRW     27$
              00000000G      EF D5 00064 5$:   TSTL    DEF_CURRENT                             ; 2220
                            00V 13 0006A       BEQL    8$
00000000G  EF 00000000G      EF D1 0006C       CMPL    DEF_CURRENT,DEF_TAIL
                            00V 12 00077       BNEQ    8$
           50 00000000G      EF D0 00079       MOVL    DEF_CURRENT,R0                          ; 2222
00000000G  EF        05      A0 D0 00080       MOVL    5(R0),DEF_TAIL
              00000000G      EF D5 00088 8$:   TSTL    DEF_CURRENT                             ; 2227
                            00V 13 0008E       BEQL    11$
```

```
                                                        J  5
EDFUTIL                                                16-Sep-1984 00:51:37    VAX-11 Pascal V2.4-277                  Page  93
V04-000                     Generated Code              5-Sep-1984 13:38:55    DISK$VMSMASTER:[EDF.SRC]EDFUTIL.PAS;1 (59)

                 00000000G  EF 00000000G  EF   D1 00090          CMPL    DEF_CURRENT,DEF_HEAD
                                         00V   12 0009B          BNEQ    11$
                         50 00000000G     EF   D0 0009D          MOVL    DEF_CURRENT,R0                       ; 2229
                 00000000G  EF        01  A0   D0 000A4          MOVL    1(R0),DEF_HEAD
                         50 00000000G     EF   D0 000AC 11$:     MOVL    DEF_CURRENT,R0                       ; 2234
                 00000000G  EF        05  A0   D0 000B3          MOVL    5(R0),DEF_PRED
                         50 00000000G     EF   D0 000BB          MOVL    DEF_CURRENT,R0                       ; 2235
                 00000000G  EF        01  A0   D0 000C2          MOVL    1(R0),DEF_SUCC
                            00000000G     EF   D5 000CA          TSTL    DEF_PRED                             ; 2237
                                         00V   13 000D0          BEQL    13$
                         50 00000000G     EF   D0 000D2          MOVL    DEF_PRED,R0                          ; 2239
                      01 A0 00000000G     EF   D0 000D9          MOVL    DEF_SUCC,1(R0)
                            00000000G     EF   D5 000E1 13$:     TSTL    DEF_SUCC                             ; 2241
                                         00V   13 000E7          BEQL    15$
                         50 00000000G     EF   D0 000E9          MOVL    DEF_SUCC,R0                          ; 2243
                      05 A0 00000000G     EF   D0 000F0          MOVL    DEF_PRED,5(R0)
                         5C 00000000G     EF   D0 000F8 15$:     MOVL    DEF_CURRENT,R12                      ; 2245
                                  11      AC   B5 000FF          TSTW    17(R12)                              ; 2249
                                         00V   15 00102          BLEQ    18$
                                  11      AC   9F 00104          PUSHAB  17(R12)                              ; 2251
                 00000000G  EF            01   FB 00107          CALLS   #1,STR$FREE1_DX
                                  09      AC   B5 0010E 18$:     TSTW    9(R12)                               ; 2253
                                         00V   15 00111          BLEQ    20$
                                  09      AC   9F 00113          PUSHAB  9(R12)                               ; 2255
                 00000000G  EF            01   FB 00116          CALLS   #1,STR$FREE1_DX
                            00000000G     EF   DD 0011D 20$:     PUSHL   DEF_CURRENT                          ; 2259
                 00000000G  EF            01   FB 00123          CALLS   #1,PAS$DISPOSE2
                            00000000G     EF   D5 0012A          TSTL    DEF_SUCC                             ; 2261
                                         00V   13 00130          BEQL    22$
                 00000000G  EF 00000000G  EF   D0 00132          MOVL    DEF_SUCC,DEF_CURRENT                 ; 2263
                                         00V   11 0013D          BRB     26$
                            00000000G     EF   D5 0013F 22$:     TSTL    DEF_PRED                             ; 2265
                                         00V   13 00145          BEQL    24$
                 00000000G  EF 00000000G  EF   D0 00147          MOVL    DEF_PRED,DEF_CURRENT                 ; 2267
                                         00V   11 00152          BRB     25$
                      0E9A  CF            00   FB 00154 24$:     CALLS   #0,NEW_IDENT_LINE                    ; 2271
                                                  00159 25$:
                                                  00159 26$:
                                               04 00159 27$:     RET                                          ; 2275

; Routine Size: 346 bytes,    Routine Base: $CODE + 0F71

                                                  00000 DELETE_PRIMARY_SECTION:                               ; 2327
                                             001C 00000          .WORD   ^M<R2,R3,R4>
                         52       04      BC   90 00002          MOVB    @4(R12),WHICHPRIMARY
                         5C       08      BC   D0 00006          MOVL    @8(R12),WHICHPRINUM
                 00000000G  EF 00000000G  EF   D0 0000A          MOVL    DEF_HEAD,DEF_CURRENT                 ; 2336
                                          53   94 00015          CLRB    DOING                                ; 2337
                                          54   94 00017          CLRB    DONE                                 ; 2338
                         50 00000000G     EF   D0 00019 1$:      MOVL    DEF_CURRENT,R0                       ; 2348
                         5C       1A      A0   D1 00020          CMPL    26(R0),WHICHPRINUM
                                         00V   12 00024          BNEQ    5$
                         50 00000000G     EF   D0 00026          MOVL    DEF_CURRENT,R0
                                          60   95 0002D          TSTB    (R0)
                                         00V   12 0002F          BNEQ    5$
                         50 00000000G     EF   D0 00031          MOVL    DEF_CURRENT,R0
                         52       19      A0   91 00038          CMPB    25(R0),WHICHPRIMARY
```

```
                                        00V  12 0003C            BNEQ     5$
                53                      01   90 0003E            MOVB     #1,DOING                                ; 2356
                00V                     53   E9 00041 5$:        BLBC     DOING,7$                                ; 2361
          0F71  CF                      00   FB 00044            CALLS    #0,DELETE_CURRENT                       ; 2363
                                        00V  11 00049            BRB      8$
          0E66  CF                      00   FB 0004B 7$:        CALLS    #0,INCR_CURRENT                         ; 2370
                          00000000G     EF   D5 00050 8$:        TSTL     DEF_CURRENT                             ; 2376
                                        00V  13 00056            BEQL     15$
                00V                     53   E9 00058            BLBC     DOING,15$                               ; 2378
                50        00000000G     EF   D0 0005B            MOVL     DEF_CURRENT,R0
                                        60   95 00062            TSTB     (R0)
                                        00V  12 00064            BNEQ     15$
                50        00000000G     EF   D0 00066            MOVL     DEF_CURRENT,R0
                5C               1A     A0   D1 0006D            CMPL     26(R0),WHICHPRINUM
                                        00V  12 00071            BNEQ     13$
                50        00000000G     EF   D0 00073            MOVL     DEF_CURRENT,R0
                52               19     A0   91 0007A            CMPB     25(R0),WHICHPRIMARY
                                        00V  13 0007E            BEQL     15$
                54                      01   90 00080 13$:       MOVB     #1,DONE                                 ; 2390
                00V                     54   E8 00083 15$:       BLBS     DONE,17$
                          00000000G     EF   D5 00086            TSTL     DEF_CURRENT
                                        8B   12 0008C            BNEQ     1$
                                             04 0008E 17$:       RET                                              ; 2394

; Routine Size: 143 bytes,    Routine Base: $CODE + 010CB

                                                00000 INIT_DEF:                                                   ; 2441
                                           0000 00000            .WORD    ^M<>
     00000000G  EF        00000000G     EF   D0 00002            MOVL     DEF_HEAD,DEF_CURRENT                    ; 2448
                                        00V  13 0000D            BEQL     6$                                      ; 2450
          0F71  CF                      00   FB 0000F 2$:        CALLS    #0,DELETE_CURRENT                       ; 2456
     00000000G  EF        00000000G     EF   D1 00014            CMPL     DEF_HEAD,DEF_TAIL
                                        EE   12 0001F            BNEQ     2$
                          00000000G     EF   D5 00021            TSTL     DEF_CURRENT                             ; 2460
                                        00V  13 00027            BEQL     6$
          0F71  CF                      00   FB 00029            CALLS    #0,DELETE_CURRENT                       ; 2462
                                             04 0002E 6$:        RET                                              ; 2466

; Routine Size: 47 bytes,    Routine Base: $CODE + 0115A

                                                00000 INSERT_IN_ORDER:                                            ; 2516
                                           001C 00000            .WORD    ^M<R2,R3,R4>
                5C               04     BC   D0 00002            MOVL     @4(R12),COLLISION_ACTION
                50        00000000G     EF   D0 00006            MOVL     DEF_SCRATCH,R0                          ; 2528
                                        60   95 0000D            TSTB     (R0)
                                        00V  12 0000F            BNEQ     3$
                50        00000000G     EF   D0 00011            MOVL     DEF_SCRATCH,R0
                0F               19     A0   91 00018            CMPB     25(R0),#15
                                        00V  13 0001C            BEQL     3$
                50        00000000G     EF   D0 0001E            MOVL     DEF_SCRATCH,R0                          ; 2534
                                 11     A0   B4 00025            CLRW     17(R0)
                50        00000000G     EF   D0 00028 3$:        MOVL     DEF_SCRATCH,R0                          ; 2536
                                 01     A0   D4 0002F            CLRL     1(R0)
                50        00000000G     EF   D0 00032            MOVL     DEF_SCRATCH,R0                          ; 2537
                                 05     A0   D4 00039            CLRL     5(R0)
                                        52   94 0003C            CLRB     BACKUP_WORKED                           ; 2542
                          00000000G     EF   D5 0003E            TSTL     DEF_CURRENT                             ; 2544
```

```
                                03  12 00044          BNEQ    .+3
                             0000V  31 00046          BRW     14$
               50 00000000G     EF  D0 00049          MOVL    DEF_CURRENT,R0                          ; 2546
                             05 A0  D5 00050          TSTL    5(R0)
                                03  12 00053          BNEQ    .+3
                             0000V  31 00055          BRW     14$
               0E80    CF       00  FB 00058          CALLS   #0,DECR_CURRENT                         ; 2550
                               00V  11 0005D          BRB     7$                                      ; 2552
               0E66    CF       00  FB 0005F  6$:     CALLS   #0,INCR_CURRENT                         ; 2560
                             01 8F  9F 00064  7$:     PUSHAB  #1
               50 00000000G     EF  D0 00067          MOVL    DEF_SCRATCH,R0
                                60  9F 0006E          PUSHAB  (R0)
               0ACF    CF       02  FB 00070          CALLS   #2,CURRENT_GT_TEST
                       53       50  90 00075          MOVB    R0,R3
                             01 8F  9F 00078          PUSHAB  #1
               50 00000000G     EF  D0 0007B          MOVL    DEF_SCRATCH,R0
                                60  9F 00082          PUSHAB  (R0)
               0C3B    CF       02  FB 00084          CALLS   #2,CURRENT_EQ_TEST
                       50       53  88 00089          BISB2   R3,R0
                                53  94 0008C          CLRB    R3
               51 00000000G     EF  D0 0008E          MOVL    DEF_CURRENT,R1
                             01 A1  D5 00095          TSTL    1(R1)
                               00V  12 00098          BNEQ    9$
                                53  96 0009A          INCB    R3
                       53       50  88 0009C  9$:     BISB2   R0,R3
                       BD       53  E9 0009F          BLBC    R3,6$
                             01 8F  9F 000A2          PUSHAB  #1                                      ; 2562
               50 00000000G     EF  D0 000A5          MOVL    DEF_SCRATCH,R0
                                60  9F 000AC          PUSHAB  (R0)
               0C3B    CF       02  FB 000AE          CALLS   #2,CURRENT_EQ_TEST
                       53       50  90 000B3          MOVB    R0,R3
                             01 8F  9F 000B6          PUSHAB  #1
               50 00000000G     EF  D0 000B9          MOVL    DEF_SCRATCH,R0
                                60  9F 000C0          PUSHAB  (R0)
               0B85    CF       02  FB 000C2          CALLS   #2,CURRENT_LT_TEST
                                51  94 000C7          CLRB    R1
               54 00000000G     EF  D0 000C9          MOVL    DEF_CURRENT,R4
                             01 A4  D5 000D0          TSTL    1(R4)
                               00V  13 000D3          BEQL    12$
                                51  96 000D5          INCB    R1
                       50       51  8A 000D7  12$:    BICB2   R1,R0
        52             53       50  89 000DA          BISB3   R0,R3,BACKUP_WORKED
                       00V      52  E8 000DE  14$:    BLBS    BACKUP_WORKED,21$                       ; 2574
         00000000G     EF 00000000G EF  D0 000E1      MOVL    DEF_HEAD,DEF_CURRENT                    ; 2581
                               00V  11 000EC          BRB     17$                                     ; 2583
               0E66    CF       00  FB 000EE  16$:    CALLS   #0,INCR_CURRENT                         ; 2591
                             01 8F  9F 000F3  17$:    PUSHAB  #1
               52 00000000G     EF  D0 000F6          MOVL    DEF_SCRATCH,R2
                                62  9F 000FD          PUSHAB  (R2)
               0ACF    CF       02  FB 000FF          CALLS   #2,CURRENT_GT_TEST
                       52       50  90 00104          MOVB    R0,R2
                             01 8F  9F 00107          PUSHAB  #1
               50 00000000G     EF  D0 0010A          MOVL    DEF_SCRATCH,R0
                                60  9F 00111          PUSHAB  (R0)
               0C3B    CF       02  FB 00113          CALLS   #2,CURRENT_EQ_TEST
                       50       52  88 00118          BISB2   R2,R0
                                52  94 0011B          CLRB    R2
```

```
                  53 00000000G   EF   D0 0011D              MOVL    DEF_CURRENT,R3
                          01     A3   D5 00124              TSTL    1(R3)
                                 00V  12 00127              BNEQ    19$
                                 52   96 00129              INCB    R2
                  52             50   88 0012B  19$:        BISB2   R0,R2
                  BD             52   E9 0012E              BLBC    R2,16$
                          01     8F   9F 00131  21$:        PUSHAB  #1                                        ; 2598
                  50 00000000G   EF   D0 00134              MOVL    DEF_SCRATCH,R0
                                 60   9F 0013B              PUSHAB  (R0)
            0ACF  CF             02   FB 0013D              CALLS   #2,CURRENT_GT_TEST
                  00V            50   E9 00142              BLBC    R0,23$
            0CB7  CF             00   FB 00145              CALLS   #0,INSERT_BEFORE_CURRENT                  ; 2600
                                 00V  11 0014A              BRB     34$
                          01     8F   9F 0014C  23$:        PUSHAB  #1                                        ; 2602
                  50 00000000G   EF   D0 0014F              MOVL    DEF_SCRATCH,R0
                                 60   9F 00156              PUSHAB  (R0)
            0C3B  CF             02   FB 00158              CALLS   #2,CURRENT_EQ_TEST
                  00V            50   E9 0015D              BLBC    R0,30$
                                 5C   D5 00160              TSTL    COLLISION_ACTION                          ; 2606
                                 00V  12 00162              BNEQ    26$
            0D30  CF             00   FB 00164              CALLS   #0,INSERT_AT_CURRENT                      ; 2608
                                 00V  11 00169              BRB     34$
                  02             5C   D1 0016B  26$:        CMPL    COLLISION_ACTION,#2                       ; 2610
                                 00V  12 0016E              BNEQ    34$
            0DED  CF             00   FB 00170              CALLS   #0,INSERT_AFTER_CURRENT                   ; 2612
                                 00V  11 00175              BRB     34$
                  50 00000000G   EF   D0 00177  30$:        MOVL    DEF_CURRENT,R0                            ; 2618
                          01     A0   D5 0017E              TSTL    1(R0)
                                 00V  12 00181              BNEQ    34$
       00000000G  EF 00000000G   EF   D0 00183              MOVL    DEF_CURRENT,DEF_TAIL                      ; 2622
            0DED  CF             00   FB 0018E              CALLS   #0,INSERT_AFTER_CURRENT                   ; 2623
                                      04 00193  34$:        RET                                               ; 2627

; Routine Size: 404 bytes,    Routine Base: $CODE + 01189

                                         00000 FIND_OBJECT:                                                   ; 2678
                                    000C 00000              .WORD   ^M<R2,R3>
                  5E        C0   AE   9E 00002              MOVAB   -64(SP),SP
                  50        04   BC   90 00006              MOVB    @4(R12),OBJ_TYP
                  51        08   BC   90 0000A              MOVB    @8(R12),PRIM
                  52        0C   BC   D0 0000E              MOVL    @12(R12),PRIMNUM
                  53        10   BC   90 00012              MOVB    @16(R12),SECO
                  5C        14   BC   D0 00016              MOVL    @20(R12),SECONUM
            C0    AD             50   90 0001A              MOVB    OBJ_TYP,TEST                              ; 2696
            D9    AD             51   90 0001E              MOVB    PRIM,TEST+25                              ; 2697
            DA    AD             52   D0 00022              MOVL    PRIMNUM,TEST+26                           ; 2698
            DE    AD             53   90 00026              MOVB    SECO,TEST+30                              ; 2699
            DF    AD             5C   D0 0002A              MOVL    SECONUM,TEST+31                           ; 2700
       00000000G  EF 00000000G   EF   D0 0002E              MOVL    DEF_HEAD,DEF_CURRENT                      ; 2705
                                 5C   94 00039              CLRB    FOUND_IT                                  ; 2706
                                 53   94 0003B              CLRB    PAST_IT                                   ; 2707
                     00000000G   EF   D5 0003D              TSTL    DEF_CURRENT                               ; 2709
                                 00V  13 00043              BEQL    8$
                          01     8F   9F 00045  2$:         PUSHAB  #1                                        ; 2715
                          C0     AD   9F 00048              PUSHAB  TEST
            0C3B  CF             02   FB 0004B              CALLS   #2,CURRENT_EQ_TEST
                  5C             50   90 00050              MOVB    R0,FOUND_IT
```

```
                                      01    8F    9F 00053              PUSHAB   #1                                              ; 2716
                                      C0    AD    9F 00056              PUSHAB   TEST
                         0ACF   CF          02    FB 00059              CALLS    #2,CURRENT_GT_TEST
                                53          50    90 0005E              MOVB     R0,PAST_IT
                                00V         5C    E8 00061              BLBS     FOUND_IT,4$                                     ; 2718
                         0E66   CF          00    FB 00064              CALLS    #0,INCR_CURRENT                                 ; 2720
                                00V         5C    E8 00069 4$:          BLBS     FOUND_IT,8$
                                00V         53    E8 0006C              BLBS     PAST_IT,8$
                                  00000000G EF    D5 0006F              TSTL     DEF_CURRENT
                                            CE    12 00075              BNEQ     2$
                                50          5C    90 00077 8$:          MOVB     FIND_OBJECT,R0                                  ; 2731
                                                  04 0007A              RET

; Routine Size: 123 bytes,    Routine Base: $CODE + 0131D

                                                0000 00000              .ENTRY   POINT_AT_DEFINITION,^M<>                        ; 2781
                    00V00000000G  EF          00    E0 00002              BBS      #0,POINTING_AT_DEFINITION,2$                    ; 2785
                       00000000G  EF 00000000G EF   D0 0000A              MOVL     DEF_HEAD,DEF_ANL_HEAD                           ; 2789
                       00000000G  EF 00000000G EF   D0 00015              MOVL     DEF_TAIL,DEF_ANL_TAIL                           ; 2790
                       00000000G  EF 00000000G EF   D0 00020              MOVL     DEF_SAVE_HEAD,DEF_HEAD                          ; 2791
                       00000000G  EF 00000000G EF   D0 0002B              MOVL     DEF_SAVE_TAIL,DEF_TAIL                          ; 2792
                       00000000G  EF          01    90 00036              MOVB     #1,POINTING_AT_DEFINITION                       ; 2794
                                                  04 0003D 2$:          RET                                                      ; 2798

; Routine Size: 62 bytes,    Routine Base: $CODE + 01398

                                                   00000 POINT_AT_ANALYSIS:                                                      ; 2848
                                                0000 00000              .WORD    ^M<>
                    00V00000000G  EF          00    E1 00002              BBC      #0,POINTING_AT_DEFINITION,2$                    ; 2852
                       00000000G  EF 00000000G EF   D0 0000A              MOVL     DEF_HEAD,DEF_SAVE_HEAD                          ; 2856
                       00000000G  EF 00000000G EF   D0 00015              MOVL     DEF_TAIL,DEF_SAVE_TAIL                          ; 2857
                       00000000G  EF 00000000G EF   D0 00020              MOVL     DEF_ANL_HEAD,DEF_HEAD                           ; 2858
                       00000000G  EF 00000000G EF   D0 0002B              MOVL     DEF_ANL_TAIL,DEF_TAIL                           ; 2859
                                     00000000G EF   94 00036              CLRB     POINTING_AT_DEFINITION                          ; 2861
                                                  04 0003C 2$:          RET                                                      ; 2865

; Routine Size: 61 bytes,    Routine Base: $CODE + 013D6

                                                0004 00000              .ENTRY   EDF$LINE_PARSED,^M<R2>                          ; 2915
                                5C          01    D0 00002              MOVL     #1,EDF$LINE_PARSED                              ; 2922
                         0AA1   CF          00    FB 00005              CALLS    #0,MAKE_SCRATCH                                 ; 2927
                                50 00000000G EF    D0 0000A              MOVL     FDL_BLOCK,R0                                    ; 2932
00000000G  EF           18      00          60    F0 00011              INSV     (R0),#0,#24,TEMP_FDL3$TYPE
                    00V00000000G  EF          05    E1 0001A              BBC      #5,TEMP_FDL3$TYPE,2$                            ; 2937
                                50 00000000G EF    D0 00022              MOVL     FDL_BLOCK,R0
                                09          08    A0    91 00029              CMPB     8(R0),#9
                                            03    12 0002D              BNEQ     .+3
                                         0000V    31 0002F              BRW      76$
                   03 00000000G  EF          03    E1 00032 2$:          BBC      #3,TEMP_FDL3$TYPE,.+3
                                         0000V    31 0003A              BRW      76$
                                52 00000000G EF    D0 0003D              MOVL     DEF_SCRATCH,R2                                  ; 2945
                    00V00000000G  EF          05    E1 00044              BBC      #5,TEMP_FDL3$TYPE,6$                            ; 2952
                                            62    94 0004C              CLRB     (R2)                                            ; 2954
                                            00V   11 0004E              BRB      7$
                                62          01    90 00050 6$:          MOVB     #1,(R2)                                         ; 2958
                    00V00000000G  EF          09    E1 00053 7$:          BBC      #9,TEMP_FDL3$TYPE,9$                            ; 2963
                                62          02    90 0005B              MOVB     #2,(R2)                                         ; 2965
```

```
EDFUTIL                                               B  6
V04-000                                      16-Sep-1984 00:51:37    VAX-11 Pascal V2.4-277                       Page  98
                        Generated Code        5-Sep-1984 13:38:55    DISK$VMSMASTER:[EDF.SRC]EDFUTIL.PAS;1 (59)

                 50 00000000G  EF    D0 0005E 9$:     MOVL    FDL_BLOCK,R0                                   ; 2970
           19    A2        08  A0    90 00065         MOVB    8(R0),25(R2)
                 50 00000000G  EF    D0 0006A         MOVL    FDL_BLOCK,R0                                   ; 2971
           1E    A2        14  A0    90 00071         MOVB    20(R0),30(R2)
   00v00000001G  EF                  00  E0 00076     BBS     #0,TEMP_FDL3$TYPE+1,11$                        ; 2976
   00v00000000G  EF                  09  E1 0007E     BBC     #9,TEMP_FDL3$TYPE,13$
      00000000G  EF 00000000G  EF    7D 00086 11$:    MOVQ    NULL_STRING,TEMP_DESCRIPTOR                    ; 2980
                 50 00000000G  EF    D0 00091         MOVL    FDL_BLOCK,R0                                   ; 2981
                           6C  A0    9F 00098         PUSHAB  108(R0)
                 51 00000000G  EF    D0 0009B         MOVL    FDL_BLOCK,R1
                           68  A1    9F 000A2         PUSHAB  104(R1)
         04C1    CF                  02  FB 000A5     CALLS   #2,CVT_QUAD_DESC
      00000000G  EF                  50  7D 000AA     MOVQ    R0,TEMP_DESCRIPTOR
                           09  A2    9F 000B1         PUSHAB  9(R2)                                          ; 2985
                    00000000G  EF    9F 000B4         PUSHAB  TEMP_DESCRIPTOR
      00000000G  EF                  02  FB 000BA     CALLS   #2,LIB$SCOPY_DXDX
   00v00000000G  EF                  05  E1 000C1 13$: BBC     #5,TEMP_FDL3$TYPE,15$                         ; 2992
                 0F        19  A2    91 000C9         CMPB    25(R2),#15
                                  00v 13 000CD         BEQL    19$
   00v00000000G  EF                  05  E0 000CF 15$: BBS     #5,TEMP_FDL3$TYPE,17$
                 21        1E  A2    91 000D7         CMPB    30(R2),#33
                                  00v 13 000DB         BEQL    19$
   00v00000000G  EF                  05  E0 000DD 17$: BBS     #5,TEMP_FDL3$TYPE,21$
                 50        1E  A2    9A 000E5         MOVZBL  30(R2),R0
                 50                  04  C4 000E9     MULL2   #4,R0
   00v00000000G  EF                  50  E1 000EC     BBC     R0,SEC_TYPE,21$
      00000000G  EF 00000000G  EF    7D 000F4 19$:    MOVQ    NULL_STRING,TEMP_DESCRIPTOR                    ; 3008
                 50 00000000G  EF    D0 000FF         MOVL    FDL_BLOCK,R0                                   ; 3009
                           64  A0    9F 00106         PUSHAB  100(R0)
                 51 00000000G  EF    D0 00109         MOVL    FDL_BLOCK,R1
                           60  A1    9F 00110         PUSHAB  96(R1)
         04C1    CF                  02  FB 00113     CALLS   #2,CVT_QUAD_DESC
      00000000G  EF                  50  7D 00118     MOVQ    R0,TEMP_DESCRIPTOR
                           11  A2    9F 0011F         PUSHAB  17(R2)                                         ; 3013
                    00000000G  EF    9F 00122         PUSHAB  TEMP_DESCRIPTOR
      00000000G  EF                  02  FB 00128     CALLS   #2,LIB$SCOPY_DXDX
                 50        19  A2    9A 0012F 21$:    MOVZBL  25(R2),R0                                      ; 3020
                 10                  50  D1 00133     CMPL    R0,#16
                                  00v 1E 00136         BGEQU   23$
    00vFFFFEB31  EF                  50  E1 00138     BBC     R0,C.AAH,23$
                 50 00000000G  EF    D0 00140         MOVL    FDL_BLOCK,R0                                   ; 3022
           1A    A2        0C  A0    D0 00147         MOVL    12(R0),26(R2)
                                  00v 11 0014C         BRB     24$
                           1A  A2    D4 0014E 23$:    CLRL    26(R2)                                         ; 3026
                 50        1E  A2    9A 00151 24$:    MOVZBL  30(R2),R0                                      ; 3028
       00000098  8F                  50  D1 00155     CMPL    R0,#152
                                  00v 1E 0015C         BGEQU   26$
    00vFFFFEB0F  EF                  50  E1 0015E     BBC     R0,C.AAI,26$
                 50 00000000G  EF    D0 00166         MOVL    FDL_BLOCK,R0                                   ; 3030
           1F    A2        18  A0    D0 0016D         MOVL    24(R0),31(R2)
                                  00v 11 00172         BRB     30$
           87    8F        1E  A2    91 00174 26$:    CMPB    30(R2),#-121                                   ; 3037
                                  00v 12 00179         BNEQ    28$
           1F    A2                  07  D0 0017B     MOVL    #7,31(R2)                                      ; 3039
                                  00v 11 0017F         BRB     29$
                           1F  A2    D4 00181 28$:    CLRL    31(R2)                                         ; 3043
                                        00184 29$:
```

```
                                                        C  6
EDFUTIL                                          16-Sep-1984 00:51:37    VAX-11 Pascal V2.4-277                    Page  99
V04-000               Generated Code              5-Sep-1984 13:38:55    DISK$VMSMASTER:[EDF.SRC]EDFUTIL.PAS;1 (59)

              59   8F          1E    A2   91 00184 30$:     CMPB     30(R2),#89                                       ; 3063
                                    00V   13 00189          BEQL     32$
              83   8F          1E    A2   91 0018B          CMPB     30(R2),#-125
                                    00V   12 00190          BNEQ     33$
                               23    A2   D4 00192 32$:     CLRL     35(R2)                                           ; 3074
                    50 00000000G     EF   D0 00195          MOVL     FDL_BLOCK,R0                                     ; 3075
              27   A2          34    A0   D0 0019C          MOVL     52(R0),39(R2)
                                  0000V   31 001A1          BRW      63$
                    50         1E    A2   9A 001A4 33$:     MOVZBL   30(R2),R0                                        ; 3086
        29    62   8F                50   8F 001A8          CASEB    R0,#98,#41
                                  0000V      001AD          .DISPL   34$
                                  0054       001AF          .DISPL   84
                                  0054       001B1          .DISPL   84
                                  0054       001B3          .DISPL   84
                                  0054       001B5          .DISPL   84
                                  0054       001B7          .DISPL   84
                                  0054       001B9          .DISPL   84
                                  0054       001BB          .DISPL   84
                                  0054       001BD          .DISPL   84
                                  0054       001BF          .DISPL   84
                                  0054       001C1          .DISPL   84
                                  0054       001C3          .DISPL   84
                                  0054       001C5          .DISPL   84
                                  0054       001C7          .DISPL   84
                                  0054       001C9          .DISPL   84
                                  0054       001CB          .DISPL   84
                                  0054       001CD          .DISPL   84
                                  0054       001CF          .DISPL   84
                                  0054       001D1          .DISPL   84
                                  0054       001D3          .DISPL   84
                                  0054       001D5          .DISPL   84
                                  0054       001D7          .DISPL   84
                                  0054       001D9          .DISPL   84
                                  0054       001DB          .DISPL   84
                                  0054       001DD          .DISPL   84
                                  0054       001DF          .DISPL   84
                                  0054       001E1          .DISPL   84
                                  0054       001E3          .DISPL   84
                                  0054       001E5          .DISPL   84
                                  0054       001E7          .DISPL   84
                                  0054       001E9          .DISPL   84
                                  0054       001EB          .DISPL   84
                                  0054       001ED          .DISPL   84
                                  0054       001EF          .DISPL   84
                                  0054       001F1          .DISPL   84
                                  0054       001F3          .DISPL   84
                                  0054       001F5          .DISPL   84
                                  0000V      001F7          .DISPL   50$
                                  0054       001F9          .DISPL   84
                                  0054       001FB          .DISPL   84
                                  0054       001FD          .DISPL   84
                                  0000V      001FF          .DISPL   40$
                                  0000V   31 00201          BRW      61$
                    50 00000000G     EF   D0 00204 34$:     MOVL     FDL_BLOCK,R0                                     ; 3090
        20          00         34    A0   CF 0020B          CASEL    52(R0),#0,#32
                                  0000V      00210          .DISPL   37$
                                  0042       00212          .DISPL   66
```

```
                            0042        00214           .DISPL  66
                            0042        00216           .DISPL  66
                            0042        00218           .DISPL  66
                            0042        0021A           .DISPL  66
                            0042        0021C           .DISPL  66
                            0042        0021E           .DISPL  66
                            0042        00220           .DISPL  66
                            0042        00222           .DISPL  66
                            0042        00224           .DISPL  66
                            0042        00226           .DISPL  66
                            0042        00228           .DISPL  66
                            0042        0022A           .DISPL  66
                            0042        0022C           .DISPL  66
                            0042        0022E           .DISPL  66
                            0000V       00230           .DISPL  36$
                            0042        00232           .DISPL  66
                            0042        00234           .DISPL  66
                            0042        00236           .DISPL  66
                            0042        00238           .DISPL  66
                            0042        0023A           .DISPL  66
                            0042        0023C           .DISPL  66
                            0042        0023E           .DISPL  66
                            0042        00240           .DISPL  66
                            0042        00242           .DISPL  66
                            0042        00244           .DISPL  66
                            0042        00246           .DISPL  66
                            0042        00248           .DISPL  66
                            0042        0024A           .DISPL  66
                            0042        0024C           .DISPL  66
                            0042        0024E           .DISPL  66
                            0000V       00250           .DISPL  35$
                              00V    11 00252           BRB     38$
               23     A2       1F    D0 00254  35$:     MOVL    #31,35(R2)                                ; 3092
                            0000V    31 00258           BRW     62$
               23     A2       1E    D0 0025B  36$:     MOVL    #30,35(R2)                                ; 3093
                            0000V    31 0025F           BRW     62$
               23     A2       1D    D0 00262  37$:     MOVL    #29,35(R2)                                ; 3094
                            0000V    31 00266           BRW     62$
                            0000V    31 00269  38$:     BRW     62$
               50 00000000G    EF    D0 0026C  40$:     MOVL    FDL_BLOCK,R0                              ; 3104
    06         00          34  A0    CF 00273           CASEL   52(R0),#0,#6
                            0000V       00278           .DISPL  45$
                            0000V       0027A           .DISPL  41$
                            0000V       0027C           .DISPL  46$
                            0000V       0027E           .DISPL  47$
                            0000V       00280           .DISPL  42$
                            0000V       00282           .DISPL  44$
                            0000V       00284           .DISPL  43$
                              00V    11 00286           BRB     48$
               23     A2       0D    D0 00288  41$:     MOVL    #13,35(R2)                                ; 3106
                              00V    11 0028C           BRB     49$
               23     A2       10    D0 0028E  42$:     MOVL    #16,35(R2)                                ; 3107
                              00V    11 00292           BRB     49$
               23     A2       12    D0 00294  43$:     MOVL    #18,35(R2)                                ; 3108
                              00V    11 00298           BRB     49$
               23     A2       11    D0 0029A  44$:     MOVL    #17,35(R2)                                ; 3109
                              00V    11 0029E           BRB     49$
```

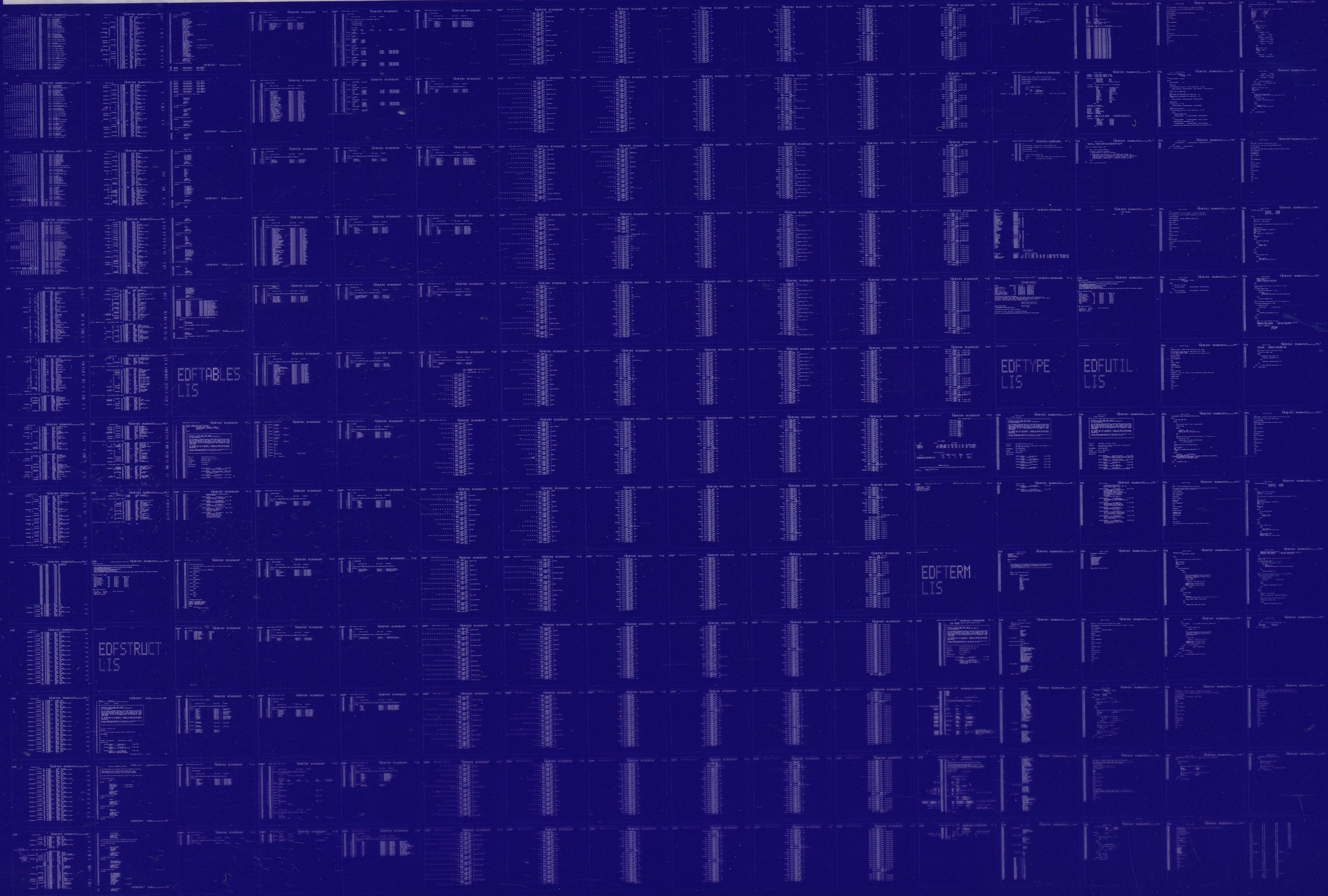
```
                   23   A2              0C   D0 002A0 45$:     MOVL     #12,35(R2)                              ; 3110
                                       00V   11 002A4          BRB      49$
                   23   A2              0E   D0 002A6 46$:     MOVL     #14,35(R2)                              ; 3111
                                       00V   11 002AA          BRB      49$
                   23   A2              0F   D0 002AC 47$:     MOVL     #15,35(R2)                              ; 3112
                                       00V   11 002B0          BRB      49$
                                                002B2 48$:
                                       00V   11 002B2 49$:     BRB      62$
                   50 00000000G         EF   D0 002B4 50$:     MOVL     FDL_BLOCK,R0                            ; 3122
        07              00        34    A0   CF 002BB          CASEL    52(R0),#0,#7
                                     0000V      002C0          .DISPL   58$
                                     0000V      002C2          .DISPL   55$
                                     0000V      002C4          .DISPL   51$
                                     0000V      002C6          .DISPL   56$
                                     0000V      002C8          .DISPL   52$
                                     0000V      002CA          .DISPL   54$
                                     0000V      002CC          .DISPL   57$
                                     0000V      002CE          .DISPL   53$
                                       00V   11 002D0          BRB      59$
                   23   A2              23   D0 002D2 51$:     MOVL     #35,35(R2)                              ; 3124
                                       00V   11 002D6          BRB      62$
                   23   A2              25   D0 002D8 52$:     MOVL     #37,35(R2)                              ; 3125
                                       00V   11 002DC          BRB      62$
                   23   A2              27   D0 002DE 53$:     MOVL     #39,35(R2)                              ; 3126
                                       00V   11 002E2          BRB      62$
                   23   A2              28   D0 002E4 54$:     MOVL     #40,35(R2)                              ; 3127
                                       00V   11 002E8          BRB      62$
                   23   A2              22   D0 002EA 55$:     MOVL     #34,35(R2)                              ; 3128
                                       00V   11 002EE          BRB      62$
                   23   A2              24   D0 002F0 56$:     MOVL     #36,35(R2)                              ; 3129
                                       00V   11 002F4          BRB      62$
                   23   A2              26   D0 002F6 57$:     MOVL     #38,35(R2)                              ; 3130
                                       00V   11 002FA          BRB      62$
                   23   A2              21   D0 002FC 58$:     MOVL     #33,35(R2)                              ; 3131
                                       00V   11 00300          BRB      62$
                                       00V   11 00302 59$:     BRB      62$
                   50 00000000G         EF   D0 00304 61$:     MOVL     FDL_BLOCK,R0                            ; 3141
                   23   A2        34    A0   D0 0030B          MOVL     52(R0),35(R2)
                   50 00000000G         EF   D0 00310 62$:     MOVL     FDL_BLOCK,R0                            ; 3145
                   27   A2        38    A0   D0 00317          MOVL     56(R0),39(R2)
                   50 00000000G         EF   D0 0031C 63$:     MOVL     FDL_BLOCK,R0                            ; 3152
                        00V       3C    A0   E9 00323          BLBC     60(R0),65$
                   2B   A2              01   90 00327          MOVB     #1,43(R2)                               ; 3154
                                       00V   11 0032B          BRB      66$
                                  2B    A2   94 0032D 65$:     CLRB     43(R2)                                  ; 3158
                   50 00000000G         EF   D0 00330 66$:     MOVL     FDL_BLOCK,R0                            ; 3160
                   2C   A2        40    A0   D0 00337          MOVL     64(R0),44(R2)
                   50 00000000G         EF   D0 0033C          MOVL     FDL_BLOCK,R0                            ; 3161
                   30   A2        48    A0   D0 00343          MOVL     72(R0),48(R2)
                   50 00000000G         EF   D0 00348          MOVL     FDL_BLOCK,R0                            ; 3162
                   34   A2        4C    A0   D0 0034F          MOVL     76(R0),52(R2)
                   50 00000000G         EF   D0 00354          MOVL     FDL_BLOCK,R0                            ; 3163
                   38   A2        50    A0   D0 0035B          MOVL     80(R0),56(R2)
                   50 00000000G         EF   D0 00360          MOVL     FDL_BLOCK,R0                            ; 3164
                   3C   A2        54    A0   D0 00367          MOVL     84(R0),60(R2)
                        50        1E    A2   9A 0036C          MOVZBL   30(R2),R0                               ; 3172
           00000098     8F              50   D1 00370          CMPL     R0,#152
```

```
                                        00V  1E 00377        BGEQU   67$
                   00VFFFFE908  EF      50   E0 00379        BBS     R0,C.AAJ,71$
                   00V00000000G EF      00   E1 00381 67$:   BBC     #0,ANALYSIS_ONLY,69$
                                50  19  A2   9A 00389        MOVZBL  25(R2),R0
                                10      50   D1 0038D        CMPL    R0,#16
                                        00V  1E 00390        BGEQU   69$
                   00VFFFFE903  EF      50   E0 00392        BBS     R0,C.AAK,71$
                   00V00000000G EF      00   E0 0039A 69$:   BBS     #0,ANALYSIS_ONLY,75$
                                50  19  A2   9A 003A2        MOVZBL  25(R2),R0
                                10      50   D1 003A6        CMPL    R0,#16
                                        00V  1E 003A9        BGEQU   71$
                   00VFFFFE8EC  EF      50   E0 003AB        BBS     R0,C.AAL,75$
                                02      62   91 003B3 71$:   CMPB    (R2),#2                                          ; 3181
                                        00V  12 003B6        BNEQ    73$
                   00000000G EF 00000000G EF D0 003B8        MOVL    DEF_TAIL,DEF_CURRENT                             ; 3185
                        0DED CF         00   FB 003C3        CALLS   #0,INSERT_AFTER_CURRENT                          ; 3186
                                        00V  11 003C8        BRB     74$
                                00000001 8F  DF 003CA 73$:   PUSHAL  #1                                               ; 3192
                        1189 CF         01   FB 003D0        CALLS   #1,INSERT_IN_ORDER
                                                003D5 74$:
                                                003D5 75$:
                                50      5C   D0 003D5 76$:   MOVL    EDF$LINE_PARSED,R0                               ; 3196
                                             04 003D8        RET

; Routine Size: 985 bytes,    Routine Base: $CODE + 01413

                                                017EC        .END
```

```
COMMAND QUALIFIERS

  PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS$:EDFUTIL/OBJ=OBJ$:EDFUTIL MSRC$:EDFUTIL

  /CHECK=(NOBOUNDS,NOCASE_SELECTORS,NOOVERFLOW,NOPOINTERS,NOSUBRANGE)
  /DEBUG=(NOSYMBOLS,NOTRACEBACK)
  /ENVIRONMENT=_$255$DUA28:[EDF.OBJ]EDFUTIL.PEN;1
  /LIST=_$255$DUA28:[EDF.LIS]EDFUTIL.LIS;1
  /OBJECT=_$255$DUA28:[EDF.OBJ]EDFUTIL.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             91       00:00.5         00:03.5
  Source Analysis           972       00:21.5         03:10.5
  Source Listing             50       00:03.9         00:07.8
  Tree Construction         134       00:01.6         00:03.5
  Flow Analysis              46       00:00.9         00:02.1
  Profit Analysis            51       00:01.2         00:02.9
  Context Analysis          479       00:13.8         00:26.7
  Name Packing                6       00:00.4         00:00.9
  Code Selection             41       00:01.9         00:03.6
  Final                     437       00:07.0         00:17.4
  TOTAL                    2311       00:52.9         04:19.1

COMPILATION STATISTICS

  CPU Time:         00:52.9          (3627 Lines/Minute)
  Elapsed Time:     04:19.1
  Page Faults:      2311
  Compilation Complete
```

0128 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



EDFTABLES
LIS

EDFTYPE
LIS

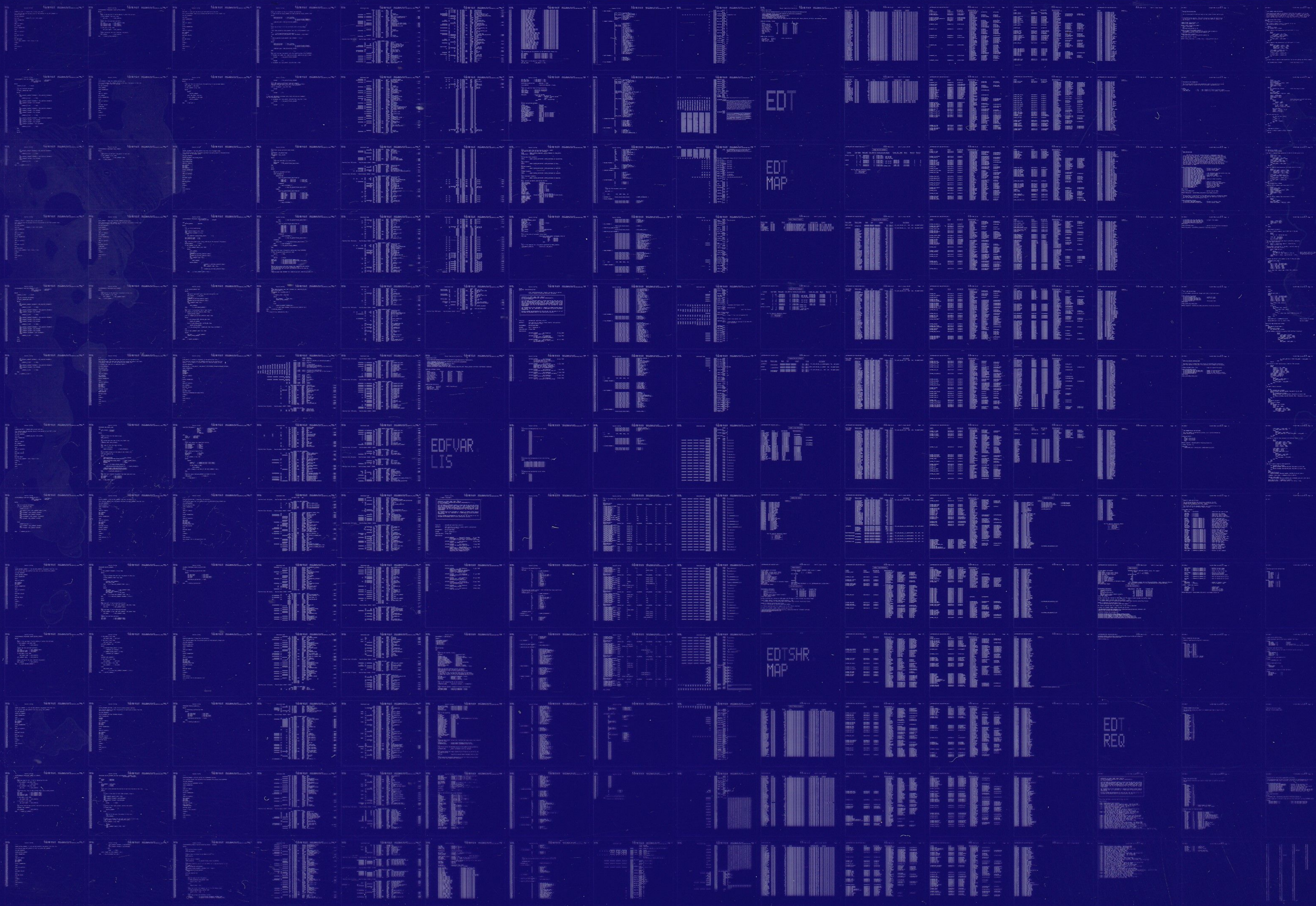
EDFUTIL
LIS

EDFTERM
LIS

EDFSTRUCT
LIS

0129 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



EDT

EDT
MAP

EDFVAR
LIS

EDTSHR
MAP

EDT
REQ