File created: 16-May-90 16:23:16 **{DSK}<usr>local>lde>lispcore>sources>EDIT.;2**

changes to: (VARS EDITCOMS)

previous date: 23-Nov-87 15:31:02 {DSK}<usr>local>lde>lispcore>sources>EDIT.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **EDITCOMS**
[;; the teletype editor
(FNS TTY/EDITE TTY/EDITL %## EDIT* EDIT%: EDITDEFAULT EDITDEFAULT1 EDITH EDITRAN EDITTO EDITXTR EDLOC
EDLOCL EDOR EDRPT EDUP ESUBST ESUBST1 EDITFERROR EDITFA EDITELT UNSAVEBLOCK? EDITF1 EDITF2 EDITL0
EDITL1 EDITL2 UNDOEDITL EDITCOM EDITCOMA EDITCOML EDITCONT EDITMAC EDITMBD EDITMV EDITCOMS
EDIT!UNDO UNDOEDITCOM UNDOEDITCOM1 EDITCOM1 EDITSAVE EDITSAVE1 EDITSMASH EDITSMASH1 EDITSW
EDITNCONC EDITAPPEND EDIT1F EDIT2F EDIT4E EDIT4E1 EDITQF EDIT4F EDIT4F1 EDIT4F2 EDIT4F3 EDITFPAT
EDITFPAT1 EDITFINDP FEDITFINDP EDITBELOW EDITBF EDITBF1 EDITNTH BPNT BPNT0 EDIT.RI EDIT.RO EDIT.LI
EDIT.LO EDIT.BI EDIT.BO)
(INITVARS (EDITRDTBL (COPYREADTABLE "OLD-INTERLISP-T")))
(USERMACROS ED)
(BLOCKS (EDITBLOCK TTY/EDITL EDITL0 EDITL1 UNDOEDITL EDITCOM EDITCOMA EDITCOML EDITMAC EDITCOMS
EDIT!UNDO UNDOEDITCOM UNDOEDITCOM1 EDITCOM1 EDITSMASH EDITSMASH1 EDITNCONC EDITAPPEND
EDIT1F EDIT2F EDITNTH BPNT BPNT0 EDIT.RI EDIT.RO EDIT.LI EDIT.LO EDIT.BI EDIT.BO
EDITDEFAULT EDITDEFAULT1 %## EDUP EDIT* EDOR EDRPT EDLOC EDLOCL EDIT%: EDITMBD EDITXTR
EDITELT EDITCONT EDITSW EDITMV EDITTO EDITBELOW EDITRAN EDITSAVE EDITSAVE1 EDITH
(ENTRIES TTY/EDITL EDITL0 %## UNDOEDITL BPNT0 EDITCONT EDLOCL)
(SPECVARS L ATM COM LCFLG %#1 %#2 %#3 UNDOLST UNDOLST1 LASTAIL MARKLST UNFIND LASTP1
LASTP2 COMS EDITCHANGES EDITHIST0 LISPXID USERHANDLE)
(RETFNS EDITL0 EDITL1)
(BLKAPPLYFNS EDIT%: EDITMBD EDITMV EDITXTR EDITSW)
(BLKLIBRARY NTH LAST MEMB NLEFT)
(NOLINKFNS PRINTDEF EDITRACEFN EDITUSERFN)
(LOCALFREEVARS FINDFLAG EDITHIST UNDOLST1 COM L L0 COM0 UNDOLST EDITLFLG ATM MARKLST
EDITHIST0 UNFIND TYPEIN LCFLG LASTP1 LASTP2 LASTAIL COPYFLG ORIGFLG COMS TOFLG C
LVL EDITCHANGES EDITLISPFLG)
(GLOBALVARS EDITCALLS P.A.STATS EDITUNDOSTATS EDITUNDOSAVES SPELLSTATS1 P.A.STATS
EDITUSERFN EDITIME DONTSAVEHISTORYCOMS COMPACTHISTORYCOMS EDITEVALSTATS MAXLOOP
EDITCOMSL EDITCOMSA DWIMFLG CLISPTRANFLG EDITOPS HISTORYCOMS REREADFLG HISTSTR3
EDITRDTBL EDITHISTORY HISTSTR0 LISPXHISTORY LISPXBUFS EDITRACEFN EDITMACROS
USERMACROS CLISPARRAY CHANGESARRAY COMMENTFLG **COMMENT**FLG EDITSMASHUSERFN))
(EDITFINDBLOCK EDIT4E EDIT4E1 EDITQF EDIT4F EDITFPAT EDITFPAT1 EDIT4F1 EDIT4F2 EDIT4F3 EDITSMASH
EDITSMASH1 EDITFINDP EDITBF EDITBF1 ESUBST (ENTRIES EDIT4E EDIT4E1 EDITQF EDIT4F EDITFPAT
EDITFINDP EDITBF ESUBST)
(LOCALFREEVARS C3 CHANGEFLG N TOPLVL FF NEWFLG FLG)
(GLOBALVARS EDITUNDOSAVES CHCONLST2 EDITQUIETFLG CHCONLST1 MAXLEVEL UPFINDFLG CLISPTRANFLG
CHANGESARRAY CLISPARRAY EDITHISTORY)
(SPECVARS ATM L COM UNFIND LASTAIL UNDOLST1 EDITCHANGES))
(NIL EDITFA TTY/EDITE (SPECVARS EDITCHANGES EDITFN)))
(GLOBALVARS FILELST DWIMFLG DWIMWAIT DWIMLOADFNSFLG)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA %##)
(NLAML EDITF2)
(LAMA])

;; the teletype editor

(DEFINEQ

(TTY/EDITE

[LAMBDA (EXPR COMS ATM TYPE IFCHANGEDFN) ; Edited 20-Nov-87 14:25 by woz

;; Used by both EDITF and EDITV. Calls EDITL in such a way that if a change occurs, and EDITL is exited via OK, STOP, or even conrol-D, the
;; appropriate call to NEWFILE? is executed. Since it checks to see if a change has been made, it also does the UNSAVEDEFING for EDITF in he
;; case that we are editing a PROP. Value is the edited expression or generates an error.

(RESETLST
(PROG ((ECHOFILE (SELECTQ (SYSTEMTYPE)
(D (TTYINFOSTREAM))
T))
EDITCHANGES TEM)
(COND
((NLISTP EXPR)
(ERROR EXPR '"not editable." T)))
[AND ATM (RESETSAVE NIL (CONS 'EDITF2 (SETQ EDITCHANGES (LIST ATM NIL TYPE IFCHANGEDFN EXPR]
(PREEDITFN ATM TYPE EDITCHANGES) ; extensions to handle editing property lists, vars etc.
[ERSETQ (SETQ TEM (COND
((SETQ EXPR (LAST (EDITL (LIST EXPR)

```
                                                        COMS ATM NIL EDITCHANGES)))
                                (CAR EXPR))
                               (T (HELP "EDITL returned NIL"]
            (COND
               ((CADR EDITCHANGES)                              ; A change was made.
                (COND
                   ((NULL TEM)
                    (ERROR!)))
                (SELECTQ TYPE
                     (FNS ;; eliminate PROP edit type.  just call PUTDEF, and let it worry about being smart about DFNFLG.
                          (PUTDEF ATM 'FNS TEM 'CHANGED))
                     (PROP (HELP "PROP edit type is no longer supported." "Where did this come from?"))
                     (VARS (SAVESET ATM TEM NIL 'NOSAVE))
                     (PROPLST (/SETPROPLIST ATM TEM))
                     NIL))
               ((NULL TEM)
                (ERROR!))
               ((EQ TYPE 'PROP)
                (PRIN1 '"not changed, so not unsaved
                       " ECHOFILE T)))
            (COND
               ((AND TYPE ATM ADDSPELLFLG)
                (ADDSPELL ATM (SELECTQ TYPE
                                  ((FNS PROP)
                                      NIL)
                                  (VARS T)
                                  (PROPLST 0)
                                  0))
                ;; TYPE is FNS or PROP for calls from EDITF, VARS for calls from EDITV, and PROPLST for calls from EDITP.  TYPE can also
                ;; be a prettytype.  Can also be the name of a CHANGEDLST in the case of a direct call from the user.

                ))
            (RETURN TEM)))])
```

(**TTY/EDITL**

```
[LAMBDA (L COMS ATM MESS EDITCHANGES)                           (* DD%: "20-Oct-81 14:02")
                                                                ; Takes edit push-down list L as argument.  Returns L as value.

  (COND
     ((NLISTP L)
      L)
     (T (PROG (LASTAIL MARKLST UNDOLST UNDOLST0 UNDOLST1 UNFIND LASTP1 LASTP2 TEM1 TEM2 EDITHIST0 EDITIME0
              EDITLISPFLG)                                      ; EDITCHANGES is a cell used for destructively marking whether
                                                                ; the edit has caused any changes.
           (COND
              ((EQ (CAR (LISTP COMS))
                   'ŸSTART)
               (SETQ READBUF (CDR COMS))
               (SETQ COMS NIL)))
           [COND
              ((AND ATM (NULL COMS)
                    EDITHISTORY)
               (SETQ EDITHIST0 T)
               (LISPXWATCH EDITCALLS)
               (SETQ EDITIME0 (CLOCK 0]
           (SETQ TEM2 (CAR (LAST L)))
      ;; TEM2 is the top level xpression.  NOte that L is usually a list of only one element, i.e. you usually start editing at the top, but not
      ;; necessarily, since editl can be called dirctly.
           [COND
              ([OR [EQ TEM2 (CAR (LAST (CAR (SETQ TEM1 (GETPROP 'EDIT 'LASTVALUE]
                   [AND ATM (EQ TEM2 (CAR (LAST (CAR (SETQ TEM1 (GETPROP ATM 'EDIT-SAVE]
                   (SOME (CAR LISPXHISTORY)
                         (FUNCTION (LAMBDA (X)
                                     (EQ TEM2 (CAR (LAST (CAR (SETQ TEM1 (CADR (FMEMB 'EDIT X]
               ;; First clause is old method of always saving last call on editor property list.  Second clause searches history list for a call to
               ;; editor corresponding to this expression.
               (AND (NULL (CDR L))
                    (SETQ L (CAR TEM1)))
               ;; if editor was called with an edit chain specified, rather just list of the xpression, use this chain.
               (SETQ MARKLST (CADR TEM1))
               (SETQ UNDOLST (CADDR TEM1))
               [COND
                  ((CAR UNDOLST)                                ; Don't want to block it twice.
                   (SETQ UNDOLST (CONS NIL UNDOLST]
               (SETQ UNDOLST0 UNDOLST)
               ;; Marks UNDOLST as of this entry to editor, so UNDO of this entire EDIT session won't go too far back.
               (SETQ UNFIND (CDDDR TEM1]
           (COND
              ((PROG1 (NLSETQ (SETQ L (EDITL0 L COMS MESS T)))
                   [COND
                      (UNDOLST1 (SETQ UNDOLST (CONS (CONS T (CONS L UNDOLST1))
                                                    UNDOLST]
```

```
                        (COND
                           ((NEQ UNDOLST UNDOLST0)
                            (AND LISPXHIST (UNDOSAVE (LIST 'UNDOEDITL L UNDOLST UNDOLST0)
                                                     LISPXHIST))    ; Takes care of making the entire call to EDITL undoable.

                            ))
                        [COND
                           (EDITIME0 (SETATOMVAL 'EDITIME (IPLUS EDITIME (IDIFFERENCE (CLOCK 0)
                                                                                      EDITIME0])
                     ;; If one of COMS causes an error, or if call to session is terminated by a STOP, still want to move undo information to
                     ;; LISPXHISTORY.
                     (RETURN L))
                    (T (ERROR!])
```

(**%##**

```
  [NLAMBDA COMS
    (PROG ((L (EVQ L))
           UNDOLST1
           (LASTAIL (EVQ LASTAIL))
           (MARKLST (EVQ MARKLST))
           (UNFIND (EVQ UNFIND)))
```

;; ## is an external entry to the editblock, so local freevariables must be looked up or traps will occur. LASAIL, MARKLT, and UNDOLST1 are ;; rebound (and therefore looked up) here to avoid their being changed by the call to ##. The rest are looked up in EDITL0 because it is called ;; with EDITLFLG=nil.

```
          (RETURN (CAR (COND
                          ((NULL COMS)
                           L)
                          (T (EDITL0 L COMS])
```

(**EDIT***

```
  [LAMBDA (N)
```

;; Equivalent to a !0 followed by an appropriate number.

```
    (CAR (SETQ L (PROG (COM (L L)
                            [X (PROG ((L L))
                                     (EDUP)
                                     (RETURN (CAR L]
                            TEM)
```

;; COM is rebound here because EDITCOM resets it so that 'CURRENT' command is typed when failure occurs. However, ;; want to see BK typed, not !0 or -3

```
                  (EDITCOM '!0)
                  (SETQ TEM (CAR L))
                  [COND
                     ([COND
                         ((MINUSP N)
                          (SETQ TEM (NLEFT TEM (MINUS N)
                                               X)))
                         (T (LISTP (SETQ TEM (CDR (NTH X N]
                      (SETQ LASTAIL TEM)
                      (RETURN (CONS (CAR TEM)
                                    L]
                  (ERROR!])
```

(**EDIT%:**

```
  [LAMBDA (TYPE LC X)                                          (* DD%: " 7-Oct-81 20:49")
    (PROG (TOFLG)
          [SETQ X (MAPCAR X (FUNCTION (LAMBDA (X)
                                      (COND
                                         [(EQ (CAR (LISTP X))
                                              '%##)
                                          (PROG ((L L)
                                                 UNDOLST1
                                                 (LCFLG T))
                                                (RETURN (COPY (EDITCOMS (CDR X]
                                         (T X]
          (COND
             (LC [COND
                    ((EQ (CAR (LISTP LC))
                         'HERE)
                     (SETQ LC (CDR LC]
                 (EDLOC LC T)))
          (EDUP)
          (SELECTQ TYPE
               ((B BEFORE)
                    (EDIT2F -1 X))
               ((A AFTER)
                    (COND
                       ((CDAR L)
                        (EDIT2F -2 X))
                       (T (EDITCOML (CONS 'N X)
                                    COPYFLG))))
```

```
            ((%: FOR)
                 [COND
                    ((OR X (CDAR L))
                     (EDIT2F 1 X))
                    ((MEMB (CAR L)
                           (CADR L))
                     ;; Singleton list, e.g. (-- ((A)) --) (DELETE A) --- result is (-- NIL --); or (-- (A) --) and say (DELETE A 1) result is (-- NIL --)
                     (EDUP)
                     (EDIT2F 1 (LIST NIL)))
                    (T                                          ; Delete last element of list of more than 1 element.
                       (EDITCOMS '(0 (NTH -2)
                                     (2])
                 (ERROR!))
          (RETURN L])
```

(EDITDEFAULT

```
  [LAMBDA (EDITX)                                               (* rmk%: " 6-JUN-82 15:13")
    (DECLARE (GLOBALVARS LPARKEY))
    (PROG (EDITY EDITZ LISPXHIST)
     ;; LISPXHIST is rebound so that messages associated with spelling corrections will not appear on history list.
          (COND
             [(AND (LISTP EDITX)
                   (SETQ EDITY (FASSOC (CAR EDITX)
                                       EDITOPS)))
              (RETURN (EDITRAN EDITX (CDR EDITY]
             [LCFLG (RETURN (COND
                              ((EQ LCFLG T)
                               (EDITQF EDITX))
                              (T                                ; E.g. LCFLG= _ in BELOW command.
                                 (EDITCOM (LIST LCFLG EDITX)
                                          TYPEIN]
             [(NLISTP EDITX)
              (COND
                 ((AND EDITHISTORY TYPEIN (FMEMB EDITX HISTORYCOMS))
                  (RETURN (EDITH EDITX)))
                 ((AND EDITUSERFN (SETQ EDITY (EDITUSERFN EDITX)))
                  (RETURN (EDITCOM EDITY TYPEIN)))
                 ((AND (NOT (U-CASEP EDITX))
                       (FMEMB (SETQ EDITY (U-CASE EDITX))
                              EDITCOMSA))
                  (SETQ EDITX EDITY)
                  (GO BACKUP))
                 ((OR (FMEMB EDITX EDITCOMSL)
                      (AND EDITY (FMEMB EDITY EDITCOMSL)
                           (SETQ EDITX EDITY)))
                  (COND
                     ((AND [NULL (CDR (SETQ EDITX (COND
                                                    (TYPEIN (READLINE EDITRDTBL (LIST EDITX)))
                                                    ((EQ EDITX (CAR COMS))
                                                     (EDITSMASH COMS (CONS (CAR COMS)
                                                                           (CDR COMS)))
                                                     (CAR COMS]
                           (NEQ (CAR EDITX)
                                '%:))
                         ;; : by itself means DELETE if nothing else follows it.  : is not an atomic command so that : -- will work as a line command.
                      (ERROR!)))
                  (AND TYPEIN (EDITSAVE1 EDITX T)))
                 ((AND TYPEIN (NULL REREADFLG)
                       (EQ LPARKEY (NTHCHAR EDITX 1)))
                  [EDITDEFAULT1 (SETQ EDITY (RPLSTRING EDITX 1 '"("]
                  (GNC EDITY)
                  [SETQ EDITX (READLINE EDITRDTBL (LIST (MKATOM EDITY]
                  (AND EDITHIST (FRPLACA (CAAAR EDITHISTORY)
                                         EDITX)))
                 ((AND TYPEIN (NULL REREADFLG)
                       (FNTYP EDITX)
                       (COND
                          ([NULL (AND (CDR (SETQ EDITY (READLINE EDITRDTBL (LIST EDITX)
                                                                 T)))
                                      (NULL (CDDR EDITY))
                                      (OR (NULL (CADR EDITY))
                                          (LISTP (CADR EDITY)))
                                      (NOT (FMEMB (CAADR EDITY)
                                                  EDITCOMSL]
                           (SETQ READBUF (APPEND (CDR EDITY)
                                                 (CONS HISTSTR0 READBUF)))
                                                                 ; put it back.
                           NIL)
                          (T T)))
                  (EDITDEFAULT1 'E EDITX)
                  (AND EDITHIST (FRPLACA (CAAR EDITHISTORY)
                                         (SETQ EDITX EDITY)))
                  (EDITH '!E)
```

```
                (RETURN))
               ([AND DWIMFLG (OR TYPEIN (EQ EDITX (CAR COMS)))
                     (SETQ EDITY (COND
                                    ((AND (EQ (NTHCHARCODE EDITX -1)
                                              (CHARCODE P))
                                          (GLC (SETQ EDITY (MKSTRING EDITX)))
                                          (SELECTQ (SETQ EDITY (MKATOM EDITY))
                                               ((^ _ UP NX BK !NX UNDO REDO CL DW)
                                                    T)
                                               (NUMBERP EDITY)))
                                                                  ; The GLC removes the last character.
                                     (EDITDEFAULT1 EDITY 'P)
                                     (CONS EDITY 'P))
                                    (T (FIXSPELL EDITX 70 EDITCOMSA (NULL TYPEIN)
                                              T]
                [COND
                   ((LISTP EDITY)
                    [COND
                       [TYPEIN (SETQ READBUF (CONS (CDR EDITY)
                                                   (CONS HISTSTR0 READBUF]
                       (T (EDITSMASH COMS (CAR EDITY)
                                 (CONS (CDR EDITY)
                                       (CDR COMS]
                    (SETQ EDITY (CAR EDITY)))
                   ((NULL TYPEIN)
                    (EDITSMASH COMS EDITY (CDR COMS]
                (SETQ EDITX EDITY)
                (GO BACKUP))
               ([AND [CDR (SETQ EDITY (COND
                                         (TYPEIN (READLINE EDITRDTBL (LIST EDITX)))
                                         ((EQ EDITX (CAR COMS))
                                          COMS]
                     (COND
                        ((NEQ (CAR EDITY)
                              EDITX)                              ; In the call to READLINE above, the user typed control-U and
                                                                  ; changed the command himself.
                         T)
                        ((AND DWIMFLG (SETQ EDITZ (FIXSPELL EDITX 70 EDITCOMSL (NULL TYPEIN)
                                                            T)))  ; E.g.  user types MBBD -- without parentheses.
                         (COND
                            [(LISTP EDITZ)
                             (EDITSMASH EDITY (CAR EDITZ)
                                    (CONS (CDR EDITZ)
                                          (CDR EDITY]
                            (T (EDITSMASH EDITY EDITZ (CDR EDITY]
                (AND (NULL TYPEIN)
                     (EDITSMASH COMS (CONS (CAR COMS)
                                           (CDR COMS)))
                     (SETQ EDITY (CAR COMS)))
                (SETQ EDITX EDITY)
                (EDITSAVE1 EDITX T))
               (T (EDITSAVE1 EDITY T)
                  (ERROR!]
          ((AND EDITHISTORY (FMEMB (CAR EDITX)
                                   HISTORYCOMS))
           (RETURN (EDITH EDITX)))
          ((AND EDITUSERFN (SETQ EDITY (EDITUSERFN EDITX)))
           (RETURN (EDITCOM EDITY TYPEIN)))
          ((NLISTP EDITX)
           (ERROR!))
          ((AND (EQ (CAR EDITX)
                    '!)
                (NULL (CDR EDITX)))
           (EDITDEFAULT1 '(1))
           (FRPLACA EDITX 1))
          ((AND (EQ (CAR EDITX)
                    '%#)
                (NULL (CDR EDITX)))
           (EDITDEFAULT1 '(3))
           (FRPLACA EDITX 3))
          [(AND DWIMFLG (ATOM (CAR EDITX))
                (SETQ EDITY (FIXSPELL (CAR EDITX)
                                      70 EDITCOMSL (NULL TYPEIN)
                                      T)))
           (COND
              [(LISTP EDITY)
               (EDITSMASH EDITX (CAR EDITY)
                      (CONS (CDR EDITY)
                            (CDR EDITX]
              (T (EDITSMASH EDITX EDITY (CDR EDITX]
          (T (ERROR!)))
      [RETURN (COND
                 ((EQ REREADFLG 'ABORT)
                  NIL)
                 (T (EDITCOM (SETQ COM EDITX)
                             TYPEIN]
BACKUP
```

```
          (SETQ COM EDITX)
          (COND
             ((AND EDITHIST TYPEIN (NULL REREADFLG))
              (FRPLACA EDITHISTORY (CDAR EDITHISTORY))
              (FRPLACA (CDR EDITHISTORY)
                       (SUB1 (CADR EDITHISTORY)))
              (EDITSAVE COM)
              ;; Can't just smash com onto front of history because now that it has been corrected, EDITSAVE may not actually save it, e.g.
              ;; suppose COM is a misspelled P.

              ))
          (RETURN (EDITCOM COM TYPEIN])
```

(EDITDEFAULT1

```
[LAMBDA (X Y)                                          ; Edited 10-Nov-87 14:06 by jds
  (PRIN1 "=" T)
  (COND
     ((STRINGP X)
      (PRIN1 X T))
     (T (PRIN2 X T T)))
  (COND
     (Y (SPACES 1 T)
        (PRIN2 Y T T)))
  (TERPRI T)
  (LISPXWATCH SPELLSTATS1])
```

(EDITH

```
[LAMBDA (C)                                            ; wt: 5-APR-77 17 56
  (PROG (X COMS LINE TEM)
        [SELECTQ C
             ((DO !E !F !N)                            ; USE is used when operator was incorrect, wheras DO is used
                                                       ; when operator was omitted.
                  [SETQ X (SELECTQ C
                              (!E                      ; !E is equivalent to DO E, !F to DO F, and !N to DO N.
                                  'E)
                              (!F 'F)
                              (!N 'N)
                              (COND
                                 ((NULL (SETQ LINE (READLINE EDITRDTBL)))
                                  (ERROR!))
                                 (T (CAR LINE]
                  (SETQ COMS (LISPXFIND EDITHISTORY NIL 'INPUT))
                  ;; If COMS is a LINE command, e.g.  FIE FUM, DO COMS is the same as (COMS FIE FUM) If COMS is a list command, e.g.
                  ;; (FIE FUM), same as (COMS (FIE FUM))
                  [COND
                     ((SETQ TEM (FMEMB HISTSTR0 COMS))
                      (COND
                         ((CDR TEM)
                          (SETQ COM C)
                          (ERROR!))
                         (T                            ; removes the last '<c.r.'
                            (SETQ COMS (LDIFF COMS TEM]
                  [SETQ COMS (COND
                                ((OR (EQ X 'E)
                                     (EQ X 'F))          ; Always a LINE command
                                 (CONS X COMS))
                                ((CDR COMS)              ; Was a LINE command.
                                 (LIST (CONS X COMS)))
                                (T                       ; Was a list command.
                                   (LIST (LIST X (CAR COMS]
                  (HISTORYSAVE EDITHISTORY '* NIL NIL COMS (LIST '*HISTORY* (CONS C LINE)))
                  (SETQ READBUF COMS)
                  (LISPXWATCH P.A.STATS))
             (UNDO (NCONC (CAAAR EDITHISTORY)
                          (SETQ LINE (READLINE EDITRDTBL)))
                   (SETQ COM NIL)
                   (SETQ X NIL)
                   [MAPC (LISPXFIND EDITHISTORY LINE 'ENTRIES T)
                         (FUNCTION (LAMBDA (Y)
                                     (AND (LISTP (SETQ Y (CADDR Y)))
                                          (SETQ X T)
                                          (UNDOEDITCOM Y T]
                   (COND
                      ((NULL X)
                       (PRIN1 '"nothing saved.
                              " T)))
                   (LISPXWATCH P.A.STATS)
                   (RETURN))
             (ŸBUFS (LISPX C)                          ; Restores input buffers.  Transparent to history.
                    (RETURN NIL))
             (RESETLST
                 (RESETSAVE (SETREADTABLE EDITRDTBL T)
                            (LIST 'SETREADTABLE (GETREADTABLE T)
                                  T))                  ; so reading and printing will be done with editreadtable.
```

```
                [RESETVARS ((LISPXHISTORY EDITHISTORY))
                           (SETQ COM NIL)
                           (RETURN (LISPX C '*])]
```

;; LISPX will set up READBUF. At this point we know C is on the list HISTORYCOMS, so it might be USE, REDO, FIX, etc. Using LISPX this ;; way means new history commands for LISPX can also be used in the editor simply by adding them to the list HISTORYCOMS.

```
      (AND READBUF (SETQ EDITHIST (CDDAAR EDITHISTORY)))     ; For saving undo information for this command (s) back in
                                                             ; EDITL1.
      (PROG (EDITHIST)
        LP  (COND
               ((NULL (SETQ READBUF (LISPXREADBUF READBUF))) ; e.g. a REDO N TIMES which just/is about to run out
                (RETURN)))
            (SETQ COM (LISPXREAD T EDITRDTBL))
            (AND EDITHISTORY (EDITSAVE COM))
            (EDITCOM COM T)
            (GO LP])
```

(EDITRAN

```
[LAMBDA (C DEF)
  (SETQ L (OR [PROG ((L L)
                     (L0 L)
                     WORDS C1 TEM)
                    (COND
                       ([AND (NULL DEF)
                             (NULL (SETQ DEF (CDR (FASSOC (CAR C)
                                                          EDITOPS]
                        (ERROR!))
                       ((NULL (SETQ WORDS (CAR DEF)))
                        (GO OUT)))
                    (COND
                       ([SETQ C1 (SOME C (FUNCTION (LAMBDA (X)
                                                   (FMEMB X WORDS]
                        (GO OUT))
                       ([SETQ C1 (SOME C (FUNCTION (LAMBDA (X Y)
                                                   (SETQ TEM (FIXSPELL X 70 WORDS (NULL TYPEIN)
                                                                       Y]
                        (EDITSMASH C1 TEM (CDR C1))
                        (GO OUT))
                       (T (ERROR!)))
                  OUT [SETQ TEM (BLKAPPLY (CAR (SETQ DEF (CADR DEF)))
                                          (PROG ((%#1 (CDR (LDIFF C C1)))
                                                 (%#2 (CAR C1))
                                                 (%#3 (CDR C1)))
                                                (RETURN (MAPCAR (CDR DEF)
                                                                (FUNCTION (LAMBDA (X)
                                                                          (COND
                                                                             ((ATOM X)
                                                                              ; So you don't have to QUOTE atoms.
                                                                              (SELECTQ X
                                                                                   (%#1 %#1)
                                                                                   (%#2 %#2)
                                                                                   (%#3 %#3)
                                                                                   X))
                                                                            (T (EVAL X]
                      (RETURN (COND
                                 ([AND TEM (CDR L0)
                                       (NOT (MEMB (CAR L0)
                                                  (CADR L0)))
                                       (NOT (TAILP (CAR L0)
                                                   (CADR L0]
                                  TEM)
                                 (T L0]
            L])
```

(EDITTO

```
[LAMBDA (LC1 LC2 FLG)                                        (* lmm "11-JUL-83 01:35")
```

;; Locates LC1 does an UP, and then attempts to do a BI at that level, i.e. LC2 specifies an element in the NTH or BI sense --- that expression at ;; this level containing C3.

```
(SETQ L (PROG ((L L))
              (COND
                 (LC1 (EDLOC LC1)
                      (EDUP)))
              (SETQ COM LC2)
              (PROG (COM)
                    (EDIT.BI 1 (COND
                                  ((AND (NUMBERP LC1)
                                        (NUMBERP LC2)
                                        (IGREATERP LC2 LC1))
                                   (IPLUS LC2 (IMINUS LC1)
                                          1))
                                  (T LC2))
                               (CAR L)))
              [COND
                 ((AND (EQ FLG 'TO)
```

```
                        (CDAAR L))                          ; Does not include endpoint.
                    (EDIT.RI 1 -2 (CAR L]
                (EDITCOM 1)
```

;; In case segment to be deleted is at beginning of list, this ensures that it is the segment that is deleted, not the list.

```
                (RETURN L)))
    (SETQ TOFLG T])
```

(EDITXTR

```
  [LAMBDA (LC X)                                            (* DD%: " 7-Oct-81 21:07")
    (PROG (TOFLG)
          (COND
             ((AND (LISTP LC)
                   (NEQ (CAR LC)
                        'HERE))
              (EDLOC LC T)))
          [PROG ([L (LIST (COND
                             ((TAILP (CAR L)
                                     (CADR L))              ; Effectively does a 1
                              (CAAR L))
                             (T (CAR L]
                 UNFIND)
                (EDLOC X T)
                (SETQ X (COND
                           ((TAILP (CAR L)
                                   (CADR L))
                            (CAAR L))
                           (T (CAR L]
          (EDUP)
          [EDIT2F 1 (COND
                       (TOFLG                               ; APPEND X for undoing.
                              (APPEND X))
                       (T (LIST X]
          [AND (NULL TOFLG)
               (LISTP (CAAR L))
               (SETQ L (CONS (CAAR L)
                             (COND
                                ((TAILP (CAR L)
                                        (CADR L))           ; To remove the extra (annoying) tail caused by the UP.
                                 (CDR L))
                                (T L]
          (RETURN L])
```

(EDLOC

```
  [LAMBDA (EDX FLG)
    (PROG ((OLDL L)
           (OLDF UNFIND)
           (LCFLG T)
           EDL FINDFLAG COMS)
          (COND
             ((NLISTP EDX)
              (EDITCOM EDX))
             ((AND (NULL (CDR EDX))
                   (NLISTP (CAR EDX)))
              (EDITCOM (CAR EDX)))
             (T (GO LP)))
          (SETQ UNFIND OLDL)
          (RETURN (CAR L))
      LP  (SETQ EDL L)
          [COND
             ((NLSETQ (EDITCOMS EDX))
              (SETQ UNFIND OLDL)
              (RETURN (CAR L]
          (COND
             ((OR FLG (EQUAL EDL L))
```

;; If command of form (LC FOO (IF --)) this will check whether failure was because there were no more FOO'S or because of the IF ;; clause. In the latter case, the search continues.

;; FLG is T on calls from EDIT:, EDITXTR, EDITMBD, and EDITMV. In this case, the search does not continue, e.g. if user says ;; (MOVE COND 3 TO AFTER --) and the next COND does not have a third clause, the MOVE fails. Of course, the user can always ;; type (MOVE (LC COND 3) TO AFTER --) if he intends to search for a COND containing three elements.

```
              (SETQ L OLDL)
              (SETQ UNFIND OLDF)
              (ERROR!)))
          (GO LP])
```

(EDLOCL

```
  [LAMBDA (COMS)
    (CAR (SETQ L (NCONC (PROG [(L (LIST (CAR L]
                              (EDLOC COMS T)
                              (RETURN L))
                        (CDR L])
```

(EDOR

```
  [LAMBDA (COMS)                                          (* lmm "22-NOV-82 00:09")
    (PROG NIL
      LP  [COND
             ((NULL COMS)
              (ERROR!))
             ([ERSETQ (SETQ L (PROG ((L L))
                                    (EDITCOMS (CAR COMS))
                                    (RETURN L]
              (RETURN (CAR L]
          (SETQ COMS (CDR COMS))
          (GO LP])
```

(EDRPT

```
  [LAMBDA (EDRX QUIET)                                    (* wt%: "14-NOV-78 02:03")
    (PROG ((EDRL L)
           (EDRPTCNT 0)
           (COPYFLG T))
      LP  (COND
             ((AND MAXLOOP (IGREATERP EDRPTCNT MAXLOOP))
              (PRIN1 '"maxloop exceeded.
                     " T))
             ((NLSETQ (RESETVARS ((MAXLOOP MAXLOOP))
                                 (EDITCOMS EDRX)))
              (SETQ EDRL L)
              (SETQ EDRPTCNT (ADD1 EDRPTCNT))
              (GO LP))
             ((NULL QUIET)
              (PRIN1 EDRPTCNT T)
              (PRIN1 '" occurrences.
                     " T)))
          (SETQ L EDRL)                                   ; L is left as of last successful completion of loop.
          (RETURN])
```

(EDUP

```
  [LAMBDA NIL
```

;; Always equivalent to a 0 followed by an appropriate NTH.

```
    (PROG (C-EXP L1 X)
          (SETQ C-EXP (CAR L))
          (COND
             ((NULL (SETQ L1 (CDR L)))
              (SETQQ COM (ERROR%: . "can't - at top.
"))
              (ERROR!))
             ((TAILP C-EXP (CAR L1))                      ; Already UP.
              (RETURN))
             ((AND (EQ C-EXP (CAR LASTAIL))
                   (TAILP LASTAIL (CAR L1)))
              (SETQ X LASTAIL))
             ([NOT (SETQ X (MEMB C-EXP (CAR L1]
              (ERROR!))
             ((MEMB C-EXP (CDR X))
              (PRIN2 C-EXP T T)
              (PRIN1 '" - location uncertain.
                     " T)
              (ERROR!)))
          [COND
             ([OR (EQ X (CAR L1))
                  (AND (EQ (CAAR L1)
                           CLISPTRANFLG)
                       (EQ X (CDDAR L1]
```

;; Since (NTH 1) is now a nop, to insure that 0 always does something, this check is to take care of 1 followed by UP.

```
              (SETQ L L1))
             (T (SETQ L (CONS X L1]
          (RETURN])
```

(ESUBST

```
  [LAMBDA (NEW OLD EXPR ERRORFLG CHARFLG)                 (* wt%: "16-FEB-79 13:08")
```

;; Does a /DSUBST a la R command in editor. Thus gives an error if Y not found in Z, and also allows you to specify X and Y using alt-modes, or ;; patterns. note that order of arguments is that of SUBST and DSUBST, not R, i.e. Y'S become X'S.

```
    (PROG ([L (LIST (SETQ EXPR (LIST EXPR]
           ATM COM UNFIND LASTAIL UNDOLST1 EDITCHANGES)
          (COND
             ((NLSETQ (EDIT4F OLD NEW T CHARFLG))
              (AND LISPXHIST (UNDOSAVE (LIST (FUNCTION ESUBST1)
                                             UNDOLST1)
                                       LISPXHIST))
              (RETURN (CAR EXPR)))
             (ERRORFLG (ERROR OLD '" ?" T)))
          (ERROR!])
```

(ESUBST1

```
[LAMBDA (X)
    ;; Undoes an ESUBST.
    (MAPC X (FUNCTION (LAMBDA (X)
                        (COND
                           ((LISTP (CAR X))
                            (/RPLNODE (CAR X)
                                   (CADR X)
                                   (CDDR X)))
                           ((EQ (CAR X)
                                'LISPXHIST)
                            ;; This is the way the editor marks an undo entry involving something other than a /rplnode, e.g. a /puthash.
                            (ESUBST1 (CDR X)))
                           (T (APPLY (CAR X)
                                     (CDR X])
```

(EDITFERROR

```
[LAMBDA (FN FLG)                                        ; Edited 17-Nov-87 16:24 by woz
    ;; called when EDITF fails to find a function.  FLG is the error message argument --- different than EDITDEF
    [if (HASDEF FN 'MACROS)
        then (PRINTOUT T "Editing macro definition for " FN T)
             (EDITDEF FN 'MACROS 'CURRENT (if (BOUNDP 'EDITCOMS)
                                              then EDITCOMS))
      elseif [AND (STRINGP FLG)
                  (OR (\DEFINEDP FN)
                      (NOT (EQ 'Y (ASKUSER DWIMWAIT 'N (CONCAT "No FNS defn for " FN ". Do you wish to edit a
                                                               dummy definition?"]
        then (ERROR FN FLG)
      else (PUTDEF FN 'FNS (EDITE (COPY DUMMY-EDIT-FUNCTION-BODY)
                                  NIL FN 'FNS]
    (AND (GETD FN)
         (if (STRINGP FLG)
             then (RETFROM 'EDITF FN)
           else FN])
```

(EDITFA

```
[LAMBDA (TYPE DEF)                                      ; Edited 10-Nov-87 14:08 by jds
    (PRIN1 '"Note: you are editing a" T)
    (AND (EQ TYPE 'ADVISED)
         (PRIN1 "n" T))
    (SPACES 1 T)
    (PRIN2 TYPE T T)
    (PRIN1 (COND
              ((EXPRP DEF)
               '" definition.")
              ((SUBRP DEF)
               '" subr!")
              (T '" compiled function!"))
           T)
    (TERPRI T])
```

(EDITELT

```
[LAMBDA (LC L)
    (PROG (Y)
          (EDLOC LC)
      LP  (SETQ Y L)
          (COND
             ((CDR (SETQ L (CDR L)))
              (GO LP)))
          (RETURN (CAR Y])
```

(UNSAVEBLOCK?

```
[LAMBDA (FN)                                            (* wt%: "27-APR-79 23:40")
    (PROG (ENTRIES)
          [MAPC FILELST (FUNCTION (LAMBDA (FILE)
                                    (MAPC
                                     (FILECOMSLST FILE 'BLOCKS)
                                     (FUNCTION (LAMBDA (BLOCK)
                                                 (AND
                                                  (CAR BLOCK)
                                                  (FMEMB FN (CDR BLOCK))
                                                  (MAPC (OR (CDR (FASSOC 'ENTRIES BLOCK))
                                                            (LIST (CAR BLOCK)))
                                                        (FUNCTION (LAMBDA (X)
                                                                    (COND
                                                                       ((AND
                                                                         (NOT (EXPRP (OR (GETPROP X 'BROKEN)
                                                                                         (GETPROP X 'ADVISED)
                                                                                         X)))
```

```
                                                                    (NOT (FMEMB X ENTRIES)))
                                                                   (SETQ ENTRIES (NCONC1 ENTRIES X]
          (COND
             (ENTRIES (MAPRINT ENTRIES T "unsave/load the definitions of the (other) entries: " " ? " ", ")
                    (COND
                       ((EQ 'Y (ASKUSER DWIMWAIT 'N NIL NIL T))
                        (MAPC ENTRIES (FUNCTION LOADDEF])
```

(EDITF1

```
  [LAMBDA (EXPR COMS ATM TYPE IFCHANGEDFN)                          (* wt%: " 8-OCT-78 19:39")
    (PRIN1 "EDITF1 has been replaced by EDITE" T)
    (EDITE EXPR COMS ATM TYPE IFCHANGEDFN])
```

(EDITF2

```
  [NLAMBDA (ATM CHANGES TYPE IFCHANGEDFN EXPR)                      (* lmm " 4-Jul-85 16:39")
    (AND CHANGES TYPE (PROG ((LISPXHIST (SELECTQ RESETSTATE
                                           ((RESET HARDRESET)
                                               NIL)
                                           LISPXHIST)))
                            (SELECTQ TYPE
                                 ((PROP FNS)
                                      (FIXEDITDATE EXPR))
                                 NIL)
                            (COND
                               (IFCHANGEDFN (APPLY* IFCHANGEDFN ATM EXPR TYPE (NULL RESETSTATE)))
                               (T (SELECTQ TYPE
                                       (PROPLST NIL)
                                       (PROP (MARKASCHANGED ATM 'FNS))
                                       (MARKASCHANGED ATM TYPE])
```

(EDITL0

```
  [LAMBDA (L COMS MESS EDITLFLG)                                    ; Edited 10-Nov-87 14:24 by jds
```

;; EDITL0 should only be called while under an EDITL since the global states of the edit, e.g. UNFIND, LASTP1, UNDOLST, etc. are all bound in ;; EDITL. Note that individual calls to EDITL0 are not undoable, i.e. any changes that are made are stored on UNDOLST or UNDOLST1, not on ;; LISPXHISTORY. Only for calls to EDITL are the changes transfered to LISPXHISTORY. Note also that when COMS are specified, all structure ;; changes are saved on UNDOLST1. When the editor is used on-line, structure changes for each command are saved on UNDOLST1 and at the ;; end of each command, gathered up and stored on UNDOLST.

```
    (PROG (FINDFLAG LCFLG TOFLG EDITHIST L0 COM0 COM COPYFLG ORIGFLG (LISPXID '*))
          (COND
             (COMS (SETQ COPYFLG T)
                   (EDITCOMS COMS)
                   (RETURN L)))
          (AND (NEQ (POSITION T)
                    0)
               (TERPRI T))
          (PRIN1 (OR MESS "edit
                          ")
                 T T)
      LP  (EDITL1)                                                  ; Only way to exit is via EDITEXIT which does a RETFROM.
          (SETQ LISPXBUFS (OR (CLBUFS T)
                              LISPXBUFS))                           ; User control-e'd out of read, CLEARBUF has already been
                                                                    ; done.
          (GO LP])
```

(EDITL1

```
  [LAMBDA (UNDOLST1 EDITHIST)                                       ; Edited 25-Nov-86 12:27 by bvm:
    (ERSETQ (PROG ((USERHANDLE 'EDITL1))
```

;; USERHANDLE mars the last place the user typed somthing to start 'computaton' started, so that if somebody wants to save state ;; and RETTO to continue computing until some condition resumes the saved state, this is the place. (If the ediotor were written to call ;; userexec and let lispx pass the edit commands to a lispxuserfn, then this wouldnt be necessary. editl1 plays the role of to editl0 that ;; lispx plays to evalqt) Thus UNDOLST1 and EDITHIST which are the only variabes associated with each event, need to be rebound ;; below EDITL!. They are rebound as arguments, even though they aret used that way, rather than puting them in as prog variabes to ;; save making an extra frame.

;; USERHANDLE appears to be obsolete now --bvm 11/86

```
          CT  (SETQ FINDFLAG NIL)
          A   (SETQ EDITHIST NIL)
              (SETQ UNDOLST1 NIL)
              (FRESHLINE T)                                         ; Holds any changes from execution of this command.
              (PROMPTCHAR '* NIL EDITHISTORY)
              (SETQ COM (LISPXREAD T EDITRDTBL))
              (SETQ L0 L)                                           ; Marks L as of beginning of this command. Used by UNDO.
              [SETQ COM0 (COND
                            ((NLISTP COM)
                             COM)
                            (T (CAR COM]                            ; Saves command name. Needed for storing on UNDOLST
                                                                    ; below.
```

;; Saves current L and command name for UNDOLST. Command name may be changed during execution to enable better error ;; diagnostics, e.g. on any find commands inside of a complicated operation.

```
              (AND EDITHISTORY (EDITSAVE COM))
              (COND
```

```
                    ((PROG1 (CL:CATCH 'EDITSTOP
                                 (XNLSETQ (EDITCOM COM T)))
                            [COND
                               (UNDOLST1 (SETQ UNDOLST1 (CONS COM0 (CONS L0 UNDOLST1)))
                                      (SETQ UNDOLST (CONS UNDOLST1 UNDOLST]
                            [COND
                               (EDITHIST                              ; Set in EDITSAVE.
                                      (FRPLACA EDITHIST UNDOLST1)
                                      (COND
                                         (EDITHIST0 (LISPXPUT '*FIRSTPRINT* (LIST 'EDITL2 ATM T)
                                                            NIL EDITHIST)
                                                (SETQ EDITHIST0 NIL])
                        (GO A)))
                  (TERPRI T)
                  (SETQ LISPXBUFS (OR (CLBUFS)
                                      LISPXBUFS))
                  [COND
                     (COM                                             ; If COM is NIL, message has already been printed.
                          (COND
                             ((EQ (CAR (LISTP COM))
                                  'ERROR%:)
                              (PRIN1 (CDR COM)
                                     T))
                             (T (PRIN2 COM T T)
                                (PRIN1 '"  ?
                                       " T)))
                          (AND EDITHIST (LISPXPUT '*ERROR* COM NIL EDITHIST]
                  (GO CT])
```

(EDITL2

```
[LAMBDA (FILE ATM FLG)                                                (* wt%: 14-MAY-76 19 1)
  ;; used for printing edit histry list
  (LISPXPRIN1 (COND
                 (FLG "{started ")
                 (T "{finished "))
         FILE)
  (LISPXPRIN2 ATM FILE)
  (LISPXPRIN1 "}
         " FILE])
```

(UNDOEDITL

```
[LAMBDA (L ULST ULST0)                                                ; Edited 10-Nov-87 14:10 by jds
  (PROG (UNDOLST1 COM EDITCHANGES)
        [MAP ULST [FUNCTION (LAMBDA (X)
                             (AND (CAR X)
                                  (UNDOEDITCOM (CAR X]
             (FUNCTION (LAMBDA (X)
                         (COND
                            ((NEQ (SETQ X (CDR X))
                                  ULST0)
                             X]
        (COND
           ((NULL UNDOLST1)
            (PRINT "UNDOEDITL Bug -- Show Development" T T)))
        (EDITSMASH ULST (CAR ULST0)
              (CDR ULST0))
        (AND LISPXHIST (UNDOSAVE [LIST 'UNDOEDITL L (LIST (CONS T (CONS L UNDOLST1]
                                 LISPXHIST])
```

(EDITCOM

```
[LAMBDA (C TYPEIN)                                                    (* wt%: "25-APR-78 11:54")
  (SETQ COM C)                                                        ; In case there is an error, user will see what command was
                                                                      ; being executed.
  (SELECTQ EDITRACEFN
       (NIL)
       ((TRACE BREAK)
            (PRIN1 '"COM = " T)
            (BPNT0 C T 1 10)
            (PRIN1 '"C-EXP = " T)
            (BPNT0 (CAR L)
                   T 1 10 (CADR L))
            [COND
               ((EQ EDITRACEFN 'BREAK)
                (APPLY 'BREAK1 (LIST NIL T C]
            (TERPRI T))
       (EDITRACEFN C))
  (COND
     [FINDFLAG (COND
                  ((EQ FINDFLAG 'BF)
                   (SETQ FINDFLAG NIL)
                   (EDITBF C))
                  (T (SETQ FINDFLAG NIL)
                     (EDITQF C]
```

```
       ((NUMBERP C)
        (SETQ L (EDIT1F C L)))
       ((ATOM C)
        (EDITCOMA C (NULL TYPEIN)))
       ((LISTP C)
        (EDITCOML C (NULL TYPEIN)))
       (T (EDITDEFAULT C)))
    (CAR L])
```

(EDITCOMA

```
  [LAMBDA (C COPYFLG)                                    ; Edited 25-Nov-86 12:33 by bvm:
    ;; Interprets atomic commands.
    (LET (TEM TEM1)
         (COND
            ([AND (NULL ORIGFLG)
                  (OR (SETQ TEM (CDR (EDITMAC C USERMACROS)))
                      (SETQ TEM (CDR (EDITMAC C EDITMACROS]
             (LET ((COPYFLG T))
                  (EDITCOMS TEM)))
            (T (SELECTQ C
                   (NIL                                  ; Nop.
                        NIL)
                   ((OK STOP SAVE)
                        [COND
                           (UNDOLST1 (SETQ UNDOLST (CONS (SETQ UNDOLST1 (CONS COM0 (CONS L0 UNDOLST1)))
                                                         UNDOLST))
                                     (COND
                                        (EDITHIST (FRPLACA EDITHIST UNDOLST1]
                        [COND
                           ((AND EDITCHANGES (CADR EDITCHANGES)
                                 (OR (NULL EDITLFLG)
                                     (EQ EDITLFLG T)))
                            ;; a call to the editor completed, and exprssion was marked as being changed.  check to make sure that it isnt the
                            ;; case that all the changes were undone, and if so, mark it not changed.
                            (PROG ((LST UNDOLST))
                              ;; looks on undolst and sees if there really were any changes made this time, e.g. they might have been undon
                              LP  (COND
                                     ((OR (NULL LST)
                                          (NULL (CAR LST)))
                                      (FRPLACA (CDR EDITCHANGES)
                                               NIL))
                                     ((SELECTQ (CAAR LST)
                                               ((UNDO !UNDO NIL)
                                                T)
                                               NIL)
                                      (SETQ LST (CDR LST))
                                      (GO LP]
                   (SELECTQ C
                        (OK [COND
                               ((OR (NULL EDITLFLG)
                                    (EQ EDITLFLG T))
                                [COND
                                   ((LITATOM ATM)
                                    (REMPROP ATM 'EDIT-SAVE]
                                [PUTPROP 'EDIT 'LASTVALUE (SETQ TEM (CONS (LAST L)
                                                                          (CONS MARKLST (CONS UNDOLST L]
                                [COND
                                   (LISPXHIST (NCONC LISPXHIST (LIST 'EDIT TEM]
                                (COND
                                   ((AND EDITHIST ATM)
                                    (LISPXPUT '*PRINT* (LIST 'EDITL2 ATM)
                                              NIL EDITHIST]
                            (RETFROM 'EDITL0 L T))
                        (STOP ;; Aborts edit session.  However all changes will have been saved for undoing on UNDOLST and/or
                              ;; UNDOLST1.
                              (RETEVAL 'EDITL0 '(ERROR!)
                                       T))
                        (SAVE                            ; Exit and save.
                              [COND
                                 ((NEQ EDITLFLG T)
                                  (ERROR '"not legal under tty:" '"" T))
                                 (ATM (PUTPROP 'EDIT 'LASTVALUE (PUTPROP ATM 'EDIT-SAVE
                                                                         (CONS L (CONS MARKLST
                                                                                       (CONS UNDOLST UNFIND]
                              (RETFROM 'EDITL0 L T))
                        (SHOULDNT)))
                   (TTY%: (SETQ COM COM0)                ; So that COM0 will be printed if TTY: is aborted via stop.
                          (COND
                             ((SETQ TEM1 (LET (UNDOLST1 UNDOLST)
                                              ;; UNDOLST1 must be protected since there may have been some changes executed in
                                              ;; this command before the TTY: was reached.
```

```
                                    [SETQ TEM (NLSETQ (EDITL0 L NIL 'tty%: 'tty%:]
                                                                    ; UNDOLST1 will be NIL because TTY: can only be exited by
                                                                    ; typing in a STOP or OK.
                                    UNDOLST))
                          (SETQ UNDOLST1 (CONS (CONS 'GROUPED TEM1)
                                               UNDOLST1))

                          ;; Note that once the TTY: command has completed operation, all of the changes executed under it are
                          ;; grouped together as being changes of the TTY: command.

                          ))
                       (COND
                          (TEM                                      ; recursive edit was not aborted
                               (SETQ L (CAR TEM)))
                          ([EVALV 'COMS (SETQ TEM (STKPOS 'EDITL0]

                           ;; If COMS is not NIL, the editor is being used as subroutine, e.g. (BREAKIN -- (AFTER TTY:)).  In this case,
                           ;; want to abort the entire call to EDITL0.

                           (RETEVAL TEM '(ERROR!)
                                  T))
                          (T                                        ; Otherwise, just abort this command, e.g. (MOVE TTY TO
                                                                    ; HERE)
                             (RELSTK TEM)
                             (CL:THROW 'EDITSTOP NIL))))
(E (COND
      (TYPEIN (LISPXWATCH EDITESTATS)
              (SETQ EDITLISPFLG T)
              (LISPX (LISPXREAD T T)
                     '* NIL NIL T))
      (LCFLG (EDITQF C))
      (T (ERROR!))))
(P (COND
      ((NEQ LASTP1 L)
       (SETQ LASTP2 LASTP1)
       (SETQ LASTP1 L)))
   (BPNT0 (CAR L)
          T 1 20 (CADR L)))
(? (COND
      ((NEQ LASTP1 L)
       (SETQ LASTP2 LASTP1)
       (SETQ LASTP1 L)))
   (BPNT0 (CAR L)
          T 100 100 (CADR L)))
((PP PPV)
     (COND
        ((NEQ LASTP1 L)
         (SETQ LASTP2 LASTP1)
         (SETQ LASTP1 L)))
     (LET ((PRETTYFLG T)
           (*READTABLE* EDITRDTBL))
          (DECLARE (SPECVARS PRETTYFLG *READTABLE*))
          (PRINTDEF (CAR L)
                 NIL
                  (NEQ C 'PPV)
                 NIL NIL T)
          (TERPRI T)))
(^ (AND (CDR L)
        (SETQ UNFIND L))
   (SETQ L (FLAST L)))
(!0 ;; Continues to do 0's until TAILP is false, i.e. takes you back to next highest left parentheses regardless of state of edit
    ;; push down list
    (COND
       ((NULL (CDR L))
        (ERROR!)))
    [PROG NIL
      LP  (SETQ L (CDR L))
          (COND
             ((TAILP (CAR L)
                     (CADR L))
              (GO LP])
(MARK (SETQ MARKLST (CONS L MARKLST)))
(UNDO (COND
         [(AND TYPEIN (LISPXREADP))                 ; Indicates that this UNDO command uses the history list.
          (COND
             (EDITHISTORY (EDITH C))
             (T (ERROR!]
          (T (EDIT!UNDO TYPEIN))))
(!UNDO (EDIT!UNDO T T))
(TEST (SETQ UNDOLST (CONS NIL UNDOLST)))
(UNBLOCK (COND
            ((SETQ TEM (FMEMB NIL UNDOLST))
             (EDITSMASH TEM (CONS NIL NIL)
                    (CDR TEM)))
            (T (PRIN1 '"not blocked.
                      " T))))
(_ (COND
      (MARKLST (AND (CDR L)
```

```
                                    (SETQ UNFIND L))
                                (SETQ L (CAR MARKLST)))
                          (T (ERROR!))))
                    (\ (COND
                          (UNFIND (SETQ C L)
                                 (SETQ L UNFIND)
                                 (AND (CDR C)
                                      (SETQ UNFIND C)))
                          (T (ERROR!))))
                    (\P (COND
                           ((AND LASTP1 (NEQ LASTP1 L))
                            (SETQ L LASTP1))
                           ((AND LASTP2 (NEQ LASTP2 L))
                            (SETQ L LASTP2))
                           (T (ERROR!))))
                    (__ (COND
                           (MARKLST (AND (CDR L)
                                         (SETQ UNFIND L))
                                    (SETQ L (CAR MARKLST))
                                    (SETQ MARKLST (CDR MARKLST)))
                           (T (ERROR!))))
                    ((F BF)
                         (COND
                            (TYPEIN (SETQ TEM (LISPXREAD T EDITRDTBL))
                                    (EDITSAVE1 TEM)
                                    (SELECTQ C
                                         (F (EDITQF TEM))
                                         (BF (EDITBF TEM))
                                         (ERROR!)))
                           ((NULL COMS)
                            (ERROR!))
                           (T (SETQ FINDFLAG C)
                              NIL)))
                    (UP (EDUP))
                    (DELETE (SETQ C '(DELETE))                    ; For undoing.
                            (EDIT%: '%:))
                    (NX (EDIT* 1))
                    (BK (EDIT* -1))
                    (!NX                                          ; Goes through a string of right parentheses to next element.
                         (SETQ L (PROG ((L L)
                                        (UF L))
                                   LP   (COND
                                           ((NULL (SETQ L (CDR L)))
                                            (ERROR!))
                                           ([NULL (CDR (FMEMB (CAR L)
                                                              (CADR L]
                                            (GO LP)))
                                        (EDITCOM 'NX)
                                        (SETQ UNFIND UF)
                                        (RETURN L))))
                    (EDITDEFAULT C])
```

(EDITCOML

```
[LAMBDA (C COPYFLG)                                               (* bvm%: "18-Nov-86 17:05")
  (PROG (C2 C3 TEM)                                               ; Handles list commands.
    LP  [SETQ C2 (CAR (LISTP (SETQ C3 (CDR C]
        [SETQ C3 (CAR (LISTP (CDR (LISTP C3]
        (COND
           ((AND LCFLG (SELECTQ C2
                             ((TO THRU THROUGH to thru through)
                                  [COND
                                     ((NULL (CDDR C))
                                      (SETQ C3 -1)
                                      (SETQ C2 'THRU]
                                  T)
                             NIL))
            (EDITTO (CAR C)
                    C3 C2)
            (RETURN))
           ((NUMBERP (CAR C))
            (EDIT2F (CAR C)
                    (CDR C))
            (RETURN))
           ((EQ C2 '|..|)
            (EDITCONT (CAR C)
                    (CDDR C)
                    'N)
            (RETURN)))
        (RETURN (COND
                    [[AND (NULL ORIGFLG)
                          (OR (SETQ TEM (EDITMAC (CAR C)
                                                 USERMACROS T))
                              (SETQ TEM (EDITMAC (CAR C)
                                                 EDITMACROS T]
                     (PROG (COPYFLG)
                           (RETURN (EDITCOMS (COND
```

```
                                        ([NOT (ATOM (SETQ C3 (CAR TEM]
                                         (SUBPAIR C3 (CDR C)
                                                (CDR TEM)
                                                T))
                                        (T (SUBST (CDR C)
                                                  C3
                                                  (CDR TEM]
                (T (SELECTQ (CAR C)
                        (S (OR C2 (ERROR!))
                           [EDITCOM1 (LIST (LIST (COND
                                                    ((OR (EQ C2 '%#1)
                                                         (EQ C2 '%#2)
                                                         (EQ C2 '%#3))
                                                     'SET)
                                                    (T 'SAVESET))
                                                 C2
                                                 (PROG ((L L)
                                                        UNFIND)
                                                       (RETURN (EDLOC (CDDR C])
                        (MARK (SET C2 L))
                        (\ (SETQ UNFIND L)
                           (SETQ L (EDITCOM1 C2 T)))
                        (R (EDIT4F C2 C3 T))
                        (R1 (EDIT4F C2 C3 1))
                        ((RC RC1)
                             (EDIT4F C2 C3 (OR (EQ (CAR C)
                                                   'RC)
                                               1)
                                   T))
                        (E (SETQ TEM (EDITCOM1 C2 T))
                           (COND
                              ((NULL (CADDR C))
                               (PRINT TEM T T)))
                           TEM)
                        (I (SETQ EDITLISPFLG T)
                           (AND TYPEIN (LISPXWATCH EDITISTATS))
                           [SETQ C (CONS (COND
                                            ((ATOM C2)
                                             C2)
                                            (T (EDITCOM1 C2 T)))
                                         (EDITCOM1 (LIST (LIST 'MAPCAR
                                                               (COND
                                                                  (TYPEIN (MAPCAR (CDDR C)
                                                                                  (FUNCTION LISPX/)))
                                                                  (T (CDDR C)))
                                                               'EVAL]
                           (SETQ COPYFLG NIL)
                           (GO LP))
                        (N (COND
                              ((NLISTP (CAR L))
                               (ERROR!)))
                           [EDITNCONC (CAR L)
                                  (COND
                                     ((NLISTP (CDR C))
                                      (CDR C))
                                     (COPYFLG (COPY (CDR C)))
                                     (T                          ; APPEND makes it much easier for EDITHISTORY.
                                        (EDITAPPEND (CDR C])
                        (P (COND
                              ((NEQ LASTP1 L)
                               (SETQ LASTP2 LASTP1)
                               (SETQ LASTP1 L)))
                           (BPNT (CDR C)))
                        (F (EDIT4F C2 C3))
                        (FS [MAPC (CDR C)
                                  (FUNCTION (LAMBDA (X)
                                                    (EDITQF (SETQ COM X])
                        (F= (EDIT4F (CONS '== C2)
                                    C3))
                        (ORF (EDIT4F (COND
                                        ((CDR (LISTP (CDR C)))
                                         (CONS '*ANY* (CDR C)))
                                        (T C2))
                                     'N))
                        (BF (EDITBF C2 C3))
                        (NTH [SETQ TEM (COND
                                          ((AND (LISTP (CAR L))
                                                (EQ (CAAR L)
                                                    CLISPTRANFLG))
                                           (CDDAR L))
                                          (T (CAR L]
                             [COND
                                ((NEQ TEM (SETQ TEM (EDITNTH TEM C2)))
                                 (SETQ L (CONS TEM L])
                        (IF                                      ; Provides for conditional editing.  Form is (if pred) or (if pred
                                                                 ; coms1 coms2)
                            (COND
```

```
        ((CAR (NLSETQ (EDITCOM1 C2 T)))
                                        ; If predicate evaluates to true then perform list of commands
         (EDITCOMS C3))
        ((CDDDR C)                      ; If false and default commands given (but may be NIL) execute
                                        ; them.
         (EDITCOMS (CADDDR C)))
        (T                              ; Otherwise generate error.  This would be used to terminate a
                                        ; LP or ORR clause.
           (ERROR!))))
(RI (EDIT.RI (CADR C)
             (CADDR C)
             (CAR L)))
(RO (EDIT.RO (CADR C)
             (CAR L)))
(LI (EDIT.LI (CADR C)
             (CAR L)))
(LO (EDIT.LO (CADR C)
             (CAR L)))
(BI (EDIT.BI (CADR C)
             (CADDR C)
             (CAR L)))
(BO (EDIT.BO (CADR C)
             (CAR L)))
(M (SETQ USERMACROS (CONS [COND
                              [(NLISTP C2)
                               (COND
                                  ((SETQ TEM (EDITMAC C2 USERMACROS))
                                   (RPLACD TEM (CDDR C))
                                   (RETURN))
                                  (T (NCONC1 EDITCOMSA C2)
                                     (CONS C2 (CONS NIL (CDDR C]
                              (T (COND
                                    ((SETQ TEM (EDITMAC (CAR C2)
                                                        USERMACROS T))
                                     (RPLACA TEM (CADDR C))
                                     (RPLACD TEM (CDDDR C))
                                     (RETURN))
                                    (T (NCONC1 EDITCOMSL (CAR C2))
                                       (CONS (CAR C2)
                                             (CDDR C]
                           USERMACROS)))
(NX (EDIT* C2))
(BK (EDIT* (IMINUS C2)))
(ORR (EDOR (CDR C)))
(MBD (EDITMBD NIL (CDR C)))
(XTR (EDITXTR NIL (CDR C)))
((THRU TO)                              ; Same as (NIL THRU C2) i.e.  starts here, does an up, and then
                                        ; a (BI 1 C2) etc.
     (EDITTO NIL C2 (CAR C)))
((A B %: AFTER BEFORE)
     (EDIT%: (CAR C)
             NIL
             (CDR C)))
(MV (EDITMV NIL (CADR C)
            (CDDR C)))
((LP LPQ)
     (EDRPT (CDR C)
            (EQ (CAR C)
                'LPQ)))
(LC (EDLOC (CDR C)))
(LCL (EDLOCL (CDR C)))
(_ (SETQ L (PROG ((L L)
                  (UF L)
                  TEM)
                 (SETQ C3 (EDITFPAT C2))
           LP    [SETQ TEM (COND
                               ((AND (LISTP (CAR L))
                                     (EQ (CAAR L)
                                         CLISPTRANFLG))
                                (CDDAR L))
                               (T (CAR L]
                 (COND
                    ((COND
                        ((ATOM C3)
                         (EQ C3 (CAR TEM)))
                        [(EQ (CAR C3)
                             'IF)
                         (CAR (NLSETQ (EDITCOM1 (CADR C3)
                                                T]
                        ((OR (EQ (CAR C3)
                                 'Ÿ)
                             (EQ (CAR C3)
                                 'ŸŸ))
                                        ; Alt-mode.
                         (EDIT4E C3 (CAR TEM)))
                        (T (EDIT4E C3 TEM)))
                     (SETQ UNFIND UF)
```

```
                                    (RETURN L))
                                   ((SETQ L (CDR L))
                                    (GO LP)))
                                  (SETQ COM C2)
                                  (ERROR!)))))
        (BELOW ;; Allows specification of new position in terms of depth below some other position.  E.g. (BELOW COND
               ;; 1) indicates cond-clause you are currently in.  (BELOW \ 2) Says 2 below UNFIND.  This is useful for
               ;; getting around in long lists, e.g. user might perform an F SELECTQ then F FOO.  To get to next
               ;; SELECTQ clause, he does (BELOW \ 1) and then Observe Observe that you can also save the depth
               ;; directly by performing (S FOO (LENGTH L)) and then USE BELOW as in (BELOW ^ FOO)
               (EDITBELOW C2 C3))
        (SW (EDITSW (CADR C)
                    (CADDR C)))
        (BIND                                   ; Makes available temporary variables for EDIT macros.  Used
                                                ; by FIX8 and FIX9 macros.
              [PROG (%#1 %#2 %#3)
                    (RETURN (EDITCOMS (CDR C])
        (COMS [MAPC (CDR C)
                    (FUNCTION (LAMBDA (X)
                                      (EDITCOM (SETQ COM (EDITCOM1 X T])
        (COMSQ (EDITCOMS (CDR C)))
        (ORIGINAL (PROG ((ORIGFLG T))
                        (EDITCOMS (CDR C))))
        (RESETVAR (RESETLST
                    (APPLY (FUNCTION RESETSAVE)
                           (LIST C2 (EDITCOM1 C3 T)))
                    (EDITCOMS (CDDDR C))))
        (EDITDEFAULT C])
```

(EDITCONT

```
[LAMBDA (LC1 LC2 FLG)                                   (* DD%: " 7-Oct-81 21:56")
```

;; E.g. (COND CONTAINING RETURN) --- equivalent to three commands: F COND followed by (LCL RETURN) followed by (_ COND) NOTE: this ;; finds INNERMOST expression, i.e. if a COND contains another COND which contains a return, (COND CONTAINING RETURN) will find the ;; inner one.

```
(SETQ L (PROG ((L L))
              (SETQ LC1 (EDITFPAT LC1))
          LP  (COND
                 ((NULL (EDIT4F LC1 FLG))
                  (ERROR!))
                 ((NULL (NLSETQ (EDLOCL LC2)))
                  (GO LP)))
          LP1 (COND
                 ((NULL (SETQ L (CDR L)))
                  (ERROR!))
                 ([COND
                     [(NLISTP LC1)
                      (EQ LC1 (CAR (LISTP (CAR L]
                     [(EQ (CAR LC1)
                          'Ÿ)
                      (EDIT4E LC1 (CAR (LISTP (CAR L]
                     (T (EDIT4E LC1 (CAR L]
                  (RETURN L)))
              (GO LP1])
```

(EDITMAC

```
[LAMBDA (C LST FLG)
  (PROG (X Y)
    LP  [COND
           ((NULL LST)
            (RETURN NIL))
           ([EQ C (CAR (SETQ X (CAR LST]
            (SETQ Y (CDR X))
            (COND
               ([COND
                   (FLG (CAR Y))
                   (T (NULL (CAR Y]
                (RETURN Y]
        (SETQ LST (CDR LST))
        (GO LP])
```

(EDITMBD

```
[LAMBDA (LC X)                                          (* lmm "26-JUL-83 20:55")
  (PROG (Y TOFLG)
        (COND
           (LC (EDLOC LC T)))
        (EDUP)
        [SETQ Y (COND
                   (TOFLG (CAAR L))
                   (T (LIST (CAAR L]
        [EDIT2F 1 (COND
                     ((NULL (FEDITFINDP X EDITEMBEDTOKEN))
                      (LIST (APPEND X Y)))
```

```
                    (T (LSUBST Y EDITEMBEDTOKEN X]
          [SETQ L (CONS (CAAR L)
                        (COND
                           ((TAILP (CAR L)
                                   (CADR L))                    ; To remove the extra (annoying) tail.
                            (CDR L))
                           (T L]
          (RETURN L])
```

(EDITMV

```
  [LAMBDA (LC OP X)
    (PROG ((L0 L)
           L1 L2 TOFLG (COM0 COM))
          (COND
             ((EQ OP 'HERE)
              (COND
                 ((NULL LC)                                     ; (MOVE TO HERE --) is the same as (MOVE -- TO HERE)
                  (SETQ LC X)
                  (SETQ X NIL)))
              (SETQ OP '%:))
             [(EQ (CAR X)
                  'HERE)
              (COND
                 ((NULL LC)                                     ; (MOVE TO AFTER HERE --) is the same as (MOVE -- TO
                                                                ; AFTER HERE)
                  (SETQ LC (CDR X))
                  (SETQ X NIL))
                 (T (SETQ X (CDR X]
             ((EQ (CAR LC)
                  'HERE)                                        ; (MOVE HERE TO AFTER --) is same as (MOVE TO AFTER --)
              (SETQ LC NIL)))
          (AND X (NEQ (CAR X)
                      'TTY%:)
               (EDLOC X T))
          (PROG ((L L0)
                 (LASTAIL LASTAIL))
                (AND LC (EDLOC LC T))
                (SETQ L1 L)                                     ; L1 will be used to delete the thing being moved.
                (EDUP)
                (SETQ L2 L))
          (AND (EQ (CAR X)
                   'TTY%:)
               (EDLOC X T))
```

;; Normally we must locate X first because LC may specify TO's or THRU's which would affect numbers in X, e.g. (MOVE (2 THRU 3) TO AFTER ;; 5) However, it is distracting to do a TTY: first and then have LC fail, so in this special case, we do LC first.

```
          (SETQ COM OP)
          (COND
             ((MEMB (CAAR L2)
                    L)
              (PRIN1 '"destination is inside expression being moved.
                     " T)
              (SETQ COM COM0)
              (ERROR!)))
          [EDITCOML (COND
                       [TOFLG (CONS OP (APPEND (CAAR L2]
                       (T (LIST OP (CAAR L2]                    ; This makes COPYFLG be bound to NIL while executing this
                                                                ; command.
          (PROG ((L L1)
                 (LASTAIL (CAR L2)))
                (EDITCOMA 'DELETE))
          [SETQ UNFIND (COND
                          ((AND LC X)                           ; (MOVE -- TO AFTER --) unfind is where you put it.
                           L)
                          ([NULL (AND (CDR L2)
                                      (NOT (MEMB (CAR L2)
                                                 (CADR L2)))
                                      (NOT (TAILP (CAR L2)
                                                  (CADR L2]     ; E.g. MOVE to --, or MOVE -- to after here.  UNFIND is where
                                                                ; the thing that was moved used to be.
                           L2)
                          (T ;; CAR of L2 is not connected to the rest of L2, e.g. occurs when you MOVE the last thing in a list.  In this case,
                             ;; make UNFIND be equivalent to doing a 0 at the place where the object that was moved used to be.
                             (CDR L2]
          (RETURN L])
```

(EDITCOMS

```
  [LAMBDA (COMS)                                                ; MAPC not used because EDITDEFAULT needs tail for spelling
                                                                ; corrections.
    (PROG NIL
      LP  [COND
             ((NLISTP COMS)
              (AND COMS (EDITCOM COMS))
```

;; Permits commands that take lists of commands as arguments, e.g. ORR, IF, etc. to be given a single atomic command.

```
          (RETURN (CAR L]
        (EDITCOM (CAR COMS))
        (SETQ COMS (CDR COMS))
        (GO LP])
```

(EDIT!UNDO

```
  [LAMBDA (PRINTFLG !UNDOFLG)                                   ; Edited 10-Nov-87 14:15 by jds
    (AND EDITHISTORY (LISPXWATCH P.A.STATS))
    (PROG ((LST UNDOLST)
           FLG)
      LP   (COND
              ((OR (NULL LST)
                   (NULL (CAR LST)))
               (GO OUT)))
           (SELECTQ (CAAR LST)
                ((NIL !UNDO UNBLOCK)
                     (GO LP1))
                (UNDO (COND
                          ((NULL !UNDOFLG)
                           (GO LP1))))
                NIL)
           (UNDOEDITCOM (CAR LST)
                  PRINTFLG)
           (COND
              ((NULL !UNDOFLG)
               (RETURN)))
           (SETQ FLG T)
      LP1  (SETQ LST (CDR LST))
           (GO LP)
      OUT  (COND
              (FLG (RETURN))
              ((CDR LST)
               (PRIN1 "blocked
                      " T T))
              (T (PRIN1 '"nothing saved.
                        " T])
```

(UNDOEDITCOM

```
  [LAMBDA (X FLG)                                               ; Edited 10-Nov-87 14:31 by jds
                                                                ; If FLG is T, name of command is printed.
    (PROG (C)
          (COND
             ((NLISTP X)
              (ERROR!))
             ((NULL (SETQ C (CAR X)))                           ; Has been undone before, but UNDO it again.
              (SETQ C 'ALREADY)
              (GO OUT))
             ([NEQ (CAR (FLAST L))
                   (CAR (FLAST (CADR X]
              ;; The expression being edited is not the one referred to by this undo command.  This can happen if you undo by using history list
              ;; outside of scope of this editing.
              (PRIN1 '"different expression.
                     " T)
              (SETQ COM NIL)
              (ERROR!)))
          (SETQ L (CADR X))
          [PROG (L)                                             ; L bound to NIL so that EDITSMASH doesnt search up it looking
                                                                ; for CLISP markers.
                (UNDOEDITCOM1 X)
                (EDITSMASH X NIL (CONS (CAR X)
                                       (CDR X]                  ; Marks it so UNDO will skip it in future.  Note that undoing this
                                                                ; UNDO will unmark it.
      OUT (AND FLG (PRIN1 (COND
                             ((NULL C)
                              "already")
                             ((NOT (NUMBERP C))
                              C)
                             (T (CONS C "(--)")))
                          T T)
               (PRIN1 '" undone.
                      " T))
          (RETURN T])
```

(UNDOEDITCOM1

```
  [LAMBDA (X)
    ;; Takes a single entry on UNDOLST, i.e. list of the form (command-name L . UNDOLST1) and maps down the UNDOLST1 portion performing the
    ;; corresonding EDITSMASHes.
    (MAPC (CDDR X)
          (FUNCTION (LAMBDA (X)
                      (COND
                         ((EQ (CAR X)
```

```
                    'GROUPED)
```

;; Used by TTY: command, which must add entire UNDOLST from subordinate call to EDITL0 to its own UNDOLST1.

```
(MAPC (CDR X)
      (FUNCTION UNDOEDITCOM1)))
((EQ (CAR X)
     'LISPXHIST)
 (EDITCOM1 (CDR X)))
(T (EDITSMASH (CAR X)
              (CADR X)
              (CDDR X))
   (LISPXWATCH EDITUNDOSTATS])
```

(EDITCOM1

```
[LAMBDA (LST FLG)
```

;; LST is a list of expressions of the form used for saving undo information on LISPXHIST, i.e. CAR of form is to be applied to CDR. EDITCOM1 ;; executes the forms and then transfers the undo information to the edit history list so that UNDO can work as an edit command. EDITCOM1 is ;; used by the S and I command, and for undoing these commands. Value is the result of last application --- used by I command to get result of ;; evaluaton back.

```
(PROG ((LISPXHIST (CONS 'SIDE (CONS (LIST 0)
                                    LISPXHIST)))
       TEM)
```

;; LISPXHIST is rebound this way so that the SIDE information doesnt get stored on the regular side slot. LISPXHIST is not completely rebound, ;; i.e. to just the side info, so that other messages etc. will still be stored on the correct entry

```
      [COND
         [FLG                                                    ; IF FLG is T, LST is a single form.
              (SETQ EDITLISPFLG T)
              (SETQ TEM (EVAL (COND
                                (TYPEIN (LISPX/ LST))
                                (T LST]
         (T (MAPC LST (FUNCTION (LAMBDA (X)
                                  (SETQ TEM (COND
                                              ((NLISTP X)
                                               TEM)
                                              ((LISTP (CAR X))
                                               (/RPLNODE (CAR X)
                                                         (CADR X)
                                                         (CDDR X)))
                                              (T (APPLY (CAR X)
                                                        (CDR X]
      (AND [SETQ LISPXHIST (CDR (LISTGET1 LISPXHIST 'SIDE]
           (SETQ UNDOLST1 (CONS (CONS 'LISPXHIST LISPXHIST)
                                UNDOLST1)))
      (RETURN TEM])
```

(EDITSAVE

```
[LAMBDA (COM)                                                    (* lmm "22-NOV-82 00:11")
                                                                 ; This function was originally included in HIST but is now in the
                                                                 ; editor for block compilation.
  (PROG (X)
        (COND
           ((FMEMB COM DONTSAVEHISTORYCOMS)
            (RETURN))
           ((AND (OR (NUMBERP COM)
                     (FMEMB COM COMPACTHISTORYCOMS))
                 (OR [NUMBERP (CAR (SETQ X (CAAAR EDITHISTORY]
                     (FMEMB (CAR X)
                            COMPACTHISTORYCOMS))
                 (NOT (FMEMB HISTSTR0 X)))                       ; CAAR is first entry, CAAAR the input.
            (NCONC1 X COM)
            (RETURN X))
           ((OR (FMEMB COM HISTORYCOMS)
                (AND (LISTP COM)
                     (FMEMB (CAR COM)
                            HISTORYCOMS)))
            (RETURN)))
        (SETQ EDITHIST (CDDR (HISTORYSAVE EDITHISTORY '* NIL COM)))
```

;; EDITHIST is bound in EDITL0. Note that it is imperative for subsequently storing the undo information to save the history entry BEFORE ;; executing the command since you cannot be sure that the first entry on EDITHISTORY corresponds to the command just completed, e.g. ;; consider a loop command with a TTY in it.

```
        (COND
           (EDITLISPFLG (SETQ EDITLISPFLG NIL)))
        (RETURN NIL])
```

(EDITSAVE1

```
[LAMBDA (X REPLACEFLG)
```

;; Used to add inputs to history event, e.g. for F commands, and for line commands typed without parentheses

```
(AND EDITHIST EDITHISTORY (PROG (TEM)
                                (COND
                                   [[OR (NULL REREADFLG)
```

```
                                        (NULL (SETQ TEM (CDR (FMEMB '*GROUP* (CADR (FMEMB HISTSTR3 REREADFLG
                                                                                                    ]
                                     (COND
                                        (REPLACEFLG (FRPLACA (CAAR EDITHISTORY)
                                                             X))
                                        (T (NCONC1 (CAAAR EDITHISTORY)
                                                   X]
                                    (T                          ; Value is the list of events in the GROUP property.
                                       (COND
                                          (REPLACEFLG (FRPLACA (CAR (LAST (CAR TEM)))
                                                               X))
                                          (T (NCONC1 (CAAR (LAST (CAR TEM)))
                                                     X])
```

(EDITSMASH

```
[LAMBDA (OLD A D)                                                   (* wt%: "12-MAY-80 21:32")
                                                                    ; ALL edit changes go through this function.
   (COND
      ((NLISTP OLD)
       (ERROR!)))
   (AND EDITSMASHUSERFN (APPLY* EDITSMASHUSERFN OLD L))
   ;; hook to enable updating a structure thatis being edited that has hash links off of it.  the PROG below is a built in example of how such a thing
   ;; might be used
   (AND EDITCHANGES (FRPLACA (CDR EDITCHANGES)
                             T))
   (SETQ UNDOLST1 (CONS (CONS OLD (CONS (CAR OLD)
                                        (CDR OLD)))
                        UNDOLST1))
   (AND EDITHISTORY (LISPXWATCH EDITUNDOSAVES))
   (FRPLACA OLD A)
   (FRPLACD OLD D)
   (PROG ((L L)
          TEM)
     LP   (COND
             ((NULL L)
              (RETURN))
             ((NLISTP (CAR L)))
             [(EQ (CAAR L)
                  CLISPTRANFLG)
              ;; Deletes CLISP translation.  NOt made part of the edit event, because of the possibility of the uer performing two changes, and then
              ;; undoing the first, which would then restore the translation, even though it no longer corresond to the untranslated and chaged
              ;; CLISP.
              (COND
                 ((LISTP (SETQ TEM (CDDAR L)))
                  (/RPLNODE (CAR L)
                         (CAR TEM)
                         (CDR TEM)))
                 (T                                                 ; CLISP used to tranlsate an atom --- e.g. QLISP does this.
                    (/RPLACA (MEMB (CAR L)
                                   (CADR L))
                             TEM]
             ((AND CLISPARRAY (GETHASH (CAR L)
                                       CLISPARRAY))
              (/PUTHASH (CAR L)
                        NIL CLISPARRAY)))
          (SETQ L (CDR L))
          (GO LP))
   OLD])
```

(EDITSMASH1

```
[LAMBDA (X)
   (AND CHANGESARRAY (PROG ((L0 L))
                         LP  (COND
                                ((NULL L0)
                                 (GO OUT))
                                ((NLISTP (CAR L0)))
                                ((GETHASH (CAR L0)
                                          CHANGESARRAY)
                                 (RETURN NIL)))
                            (SETQ L0 (CDR L0))
                            (GO LP)
                         OUT [AND (NLISTP X)
                                  (SETQ X (COND
                                             ((OR (NULL (SETQ X (CADR L)))
                                                  (FMEMB (CAR L)
                                                         X))
                                              (CAR L))
                                             (T X]
                            (SETQ UNDOLST1 (CONS (CONS 'LISPXHIST (LIST (LIST '/PUTHASH X (GETHASH X
                                                                                                  CHANGESARRAY)
                                                                                          CHANGESARRAY)))
                                                 UNDOLST1))         ; Done this way for efficiency rather than going through editcom1
                                                                    ; since we know what to undosave.
```

```
                            (PUTHASH X ATM CHANGESARRAY)
                            (RETURN])
```

(EDITSW

```
  [LAMBDA (M N)
    (PROG ((Y (EDITNTH (CAR L)
                       M))
           (Z (EDITNTH (CAR L)
                       N))
           TEM)
          (SETQ TEM (CAR Y))
          (EDITSMASH Y (CAR Z)
                 (CDR Y))
          (EDITSMASH1 (CAR Z))
          (EDITSMASH Z TEM (CDR Z))
          (EDITSMASH1 TEM])
```

(EDITNCONC

```
  [LAMBDA (X Y)
    (COND
       ((NULL X)
        Y)
       ((NLISTP X)
        (ERROR!))
       (T (PROG1 X
                 (EDITSMASH (SETQ X (LAST X))
                        (CAR X)
                        Y)
                 (AND CHANGESARRAY (MAPC Y (FUNCTION EDITSMASH1)))))])
```

(EDITAPPEND

```
  [LAMBDA (X)                                                    (* wt%: " 3-OCT-78 19:59")
```

;; copies top level, differs fro append in that if ends in non-nil, the non-nil is retained

```
    (COND
       ((NLISTP X)
        X)
       (T (CONS (CAR X)
                (EDITAPPEND (CDR X])
```

(EDIT1F

```
  [LAMBDA (C L)                                                  (* wt%: "13-JUN-78 00:55")
    (PROG (TEM)
          [COND
             [(EQ C 0)
              (RETURN (COND
                          ((CDR L)
                           (RETURN (CDR L)))
                          (T (SETQQ COM (ERROR%: . "can't - at top.
"))
                             (ERROR!]
             ((NLISTP (CAR L))
              (ERROR!))
             ((EQ (CAAR L)
                  CLISPTRANFLG)
              (SETQ TEM (CDDAR L)))
             (T (SETQ TEM (CAR L]
          (RETURN (COND
                      [(IGREATERP C 0)
                       (COND
                          ((NLISTP (SETQ TEM (NTH TEM C)))
                           (ERROR!))
                          (T (CONS (CAR (SETQ LASTAIL TEM))
                                   L]
                      ([NULL (SETQ TEM (NLEFT TEM (IMINUS C]
                       (ERROR!))
                      (T (CONS (CAR (SETQ LASTAIL TEM))
                               L])
```

(EDIT2F

```
  [LAMBDA (N X)
    (PROG ([CL (COND
                  ((AND (LISTP (CAR L))
                        (EQ (CAAR L)
                            CLISPTRANFLG))
                   (CDDAR L))
                  (T (CAR L]
           TEM)
```

;; Handles all deletion, replacement and insertion. For deletion and replacement, saves information about what was destroyed on variable
;; LASTCHANGE. The command UNDO can then be used to restore the structure.

```
          [COND
```

```
            ((NLISTP CL)
             (ERROR!))
            (COPYFLG (SETQ X (COPY X)))
            (T                                                    ; APPEND makes it much easier for EDITHISTORY.
               (SETQ X (APPEND X]
         (COND
            [(IGREATERP N 0)
             (COND
                ([AND (NEQ N 1)
                      (OR [NLISTP (SETQ TEM (NTH CL (SUB1 N]
                          (NLISTP (CDR TEM]
                 (SETQ COM N)
                 (ERROR!))
                ((NULL X)                                         ; Delete
                 (GO DELETE))
                (T                                                ; Replace
                   (GO REPLACE]
            ([OR (EQ N 0)
                 (NULL X)
                 (NLISTP (SETQ TEM (NTH CL (IMINUS N]
             (ERROR!))
            (T                                                    ; Insert
               (COND
                  ((NEQ N -1)
                   (SETQ CL TEM)))                                ; Insertion also physically changes indicated tail.
               (EDITSMASH CL (CAR X)
                      (CONS (CAR CL)
                            (CDR CL)))
               (EDITSMASH1 (CAR X))
               [COND
                  ((CDR X)
                   (AND CHANGESARRAY (MAPC (CDR X)
                                           (FUNCTION EDITSMASH1)))
                   (EDITSMASH CL (CAR CL)
                          (NCONC (CDR X)
                                 (CDR CL]
               (RETURN)))
      DELETE
         [COND
            [(EQ N 1)
             (OR (LISTP (CDR CL))
                 (ERROR!))
             ;; To delete first element you must effectively replace it by second element and delete second element.  This is why you cannot delete
             ;; the first element of a list when it is the only one.
             (EDITSMASH CL (CADR CL)
                    (CDDR CL))
             (EDITSMASH1 (COND
                            ((TAILP CL (CADR L))
                             (CADR L))
                            (T CL]
            (T ;; Deleting any other element is done by patching around it, i.e.  by changing previous CDR to point to its CDR.  In general, you
               ;; can't solve problem so pointers into tails will always be updated without going down the entire list and moving everything over.
               ;; See manual.
               (EDITSMASH TEM (CAR TEM)
                      (CDDR TEM))
               (EDITSMASH1 (COND
                              ((TAILP CL (CADR L))
                               (CADR L))
                              (T CL]
         (RETURN)
      REPLACE
         [COND
            ((NEQ N 1)
             (SETQ CL (CDR TEM]
      ;; Replacement physically changes indicated tail i.e.  if you are editing (A B C D) and set FOO to (NTH 3) i.e. (C D) and then do a (3 X Y) FOO will
      ;; be changed to (X Y D)
         (EDITSMASH CL (CAR X)
                (CDR CL))
         (EDITSMASH1 (CAR X))
         (COND
            ((CDR X)
             (AND CHANGESARRAY (MAPC (CDR X)
                                     (FUNCTION EDITSMASH1)))
             (EDITSMASH CL (CAR CL)
                    (NCONC (CDR X)
                           (CDR CL])
```

(EDIT4E

```
  [LAMBDA (PAT X CHANGEFLG)                                       (* DD%: "29-MAR-83 18:02")
    (COND
       ((EQ PAT X)
        T)
       ((NLISTP PAT)
```

```
        (OR (EQ PAT '&)
            (AND (NUMBERP PAT)
                 (EQP PAT X))
            (AND (STRINGP PAT)
                 (STREQUAL PAT X)
                 T)))
       ((EQ (CAR PAT)
            '*ANY*)
        (PROG NIL
          LP  (COND
                 ((NULL (SETQ PAT (CDR PAT)))
                  (RETURN NIL))
                 ((EDIT4E (CAR PAT)
                          X)
                  (RETURN T)))
              (GO LP)))
       ((EQ (CAR PAT)
            'Ÿ)                                          ; Ÿ is the way the line printer prints alt-modes.
        (AND (OR (LITATOM X)
                 (STRINGP X))
             (EDIT4E1 (CDR PAT)
                    (DUNPACK X CHCONLST2)
                    X CHANGEFLG)))
       ((EQ (CAR PAT)
            'ŸŸ)
        ;; This pattern specifies a search for a 'close' word, using the spelling corrector, i.e. SKOR.  CADR of PAT is the number of characters in the
        ;; word, CDDR its CHCON.  The pattern is constructed by EDITFPAT when it encounters a word or string that ends in ŸŸ.
        (AND (OR (LITATOM X)
                 (STRINGP X))
             (SKOR0 X (CADR PAT)
                    (CADDR PAT)
                    (CDDDR PAT))
             (PROGN (AND (NEQ EDITQUIETFLG T)
                         (PRIN1 '= T)
                         (PRINT X T T))
                    T)))
       [(EQ (CAR PAT)
            '--)
        (OR (NULL (SETQ PAT (CDR PAT)))
            (PROG NIL
              LP  (COND
                     ((EDIT4E PAT X)
                      (RETURN T))
                     ((NLISTP X)
                      (RETURN NIL)))
                  (SETQ X (CDR X))
                  (GO LP]
       ((EQ (CAR PAT)
            '==)
        (EQ (CDR PAT)
            X))
       ((EQ (CAR (LISTP (CDR PAT)))
            '|..|)
        (AND (EDIT4E (CAR PAT)
                     (CAR X))
             [NLSETQ (PROG ((L (LIST X))
                            UNFIND ORIGFLG LASTAIL)
                           (EDLOCL (CDDR PAT]
             T))
       ((EQ (CAR PAT)
            '@)
        (APPLY* (CADR PAT)
                X))
       ((NLISTP X)
        NIL)
       ([EDIT4E (CAR PAT)
                (CAR (COND
                        ((EQ (CAR X)
                             CLISPTRANFLG)
                         (SETQ X (CDDR X)))
                        (T X]
        (EDIT4E (CDR PAT)
                (CDR X])
```

(EDIT4E1

```
  [LAMBDA (PAT LST X CHANGEFLG)
    ;; Compares PAT and X.  PAT is a DUNPACK of an atom or string which contains one or more alt-modes.  An alt-mode can match any number
    ;; (including zero) of characters in X, e.g. NUM$, $BERP, and $U$E$ all match NUMBERP.  If CHANGEFLG is T and PAT matches X, the value of
    ;; EDIT4E1 is a list of pointer pairs corresponding to the beginning and end of the sequence matched by each alt-mode.
    (PROG (PAT1 LST1 LST2 MATCH)
      LP  (COND
             [(NULL PAT)
              (COND
                 ((OR (NULL LST)
```

```
                    (NULL PAT1))
                ;; If LST is NIL, then the final characters in PAT matched those in X, e.g. $BERP vs NUMBERP.  If PAT1 is NIL, then the last
                ;; character in PAT was an altmode, e.g. NUM$ vs NUMBERP, so extra characters in LST are acceptable.
                 (GO SUCC))
               (LST1 (SETQ LST LST1)
                     (SETQ LST1 NIL)
                     (SETQ PAT PAT1))
               (T (RETURN NIL]
          ((EQ (CAR PAT)
               'Ÿ)
           [COND
              ((AND CHANGEFLG LST2 LST1)
                ;; An alt-mode was seen before.  (Note that we cannot determine the scope of an alt-mode until the next one is encountered, or
                ;; the end of the match is reached.) LST2 was the value of LST as of the beginning of the alt-mode match, LST1 the value of
                ;; LST as of its end.  However, if LST1 is NIL, then there were two alt-modes in a row, and we ignore the last one.
                 (SETQ MATCH (CONS (CONS LST2 LST1)
                                   MATCH]
           (SETQ PAT (SETQ PAT1 (CDR PAT)))
           ;; PAT1 is a pointer into PAT as of the first character after an alt-mode.  It is used for backing up after a partially successful match,
           ;; e.g. if PAT is $XYZ$ and X is XYXYZ.
           (SETQ LST1 NIL)
           (SETQ LST2 LST)
           (GO LP))
          ((NULL LST)
           (RETURN NIL))
          ((EQ (CAR PAT)
               (CAR LST))
           (COND
              ((NULL LST1)
               (SETQ LST1 LST)))
           (SETQ PAT (CDR PAT)))
          ((NULL (SETQ PAT PAT1))
           (RETURN NIL))
          (LST1 (SETQ LST LST1)
                (SETQ LST1 NIL)))
      (SETQ LST (CDR LST))
      (GO LP)
  SUCC
      (COND
         [CHANGEFLG (AND (NEQ EDITQUIETFLG T)
                         (PRIN2 X T T))                    ; EDIT4F2 will be called, and it will print -> followed by the new
                                                           ; atom or string.
                (RETURN (DREVERSE (CONS (CONS LST2 LST1)
                                        MATCH]
         ((NEQ EDITQUIETFLG T)
          (PRIN1 '= T)
          (PRINT X T T)))
      (RETURN T])
```

(**EDITQF**

```
[LAMBDA (PAT)
  (PROG (Q1)
        (COND
           ([AND (LISTP (SETQ Q1 (CAR L)))
                 (SETQ Q1 (MEMB PAT (COND
                                       ((EQ (CAR Q1)
                                            CLISPTRANFLG)
                                        (CDDDR Q1))
                                       (T (CDR Q1]
            (SETQ L (CONS (COND
                             (UPFINDFLG Q1)
                             (T (SETQ LASTAIL Q1)
                                (CAR Q1)))
                          L)))
           (T (**EDIT4F** PAT 'N])
```

(**EDIT4F**

```
[LAMBDA (PAT C3 CHANGEFLG CHARFLG)                         ; Edited 10-Nov-87 14:36 by jds
```

;; Searches the expression being edited, starting from current point and continuing in print order, until a position is found for which the current level ;; list matches PAT. Then, if (CAR L) is atomic, effectively does an UP (unless UPFINDFLG=NIL) Thus F (SETQ X --) and F SETQ will produce the ;; same result. --- If C3 is T, the search starts with the current exppession. If C3 is 'N', the search skips the current expression, although it does ;; search inside of it.

```
  (PROG (LL X TAIL (FF (CONS))
           (TOPLVL (NULL C3))
           N NEWFLG (PAT0 PAT))
        [COND
           ((EQ [CAR (LISTP (CDR (LISTP PAT]
                '|..|)
            (RETURN (**EDITCONT** (CAR PAT)
                          (CDDR PAT)
                          C3]
```

```
          (SETQ PAT (EDITFPAT PAT T))                          ; Checks PAT for altmodes.
          (SETQ LL L)
          (COND
             (CHANGEFLG (SETQ N (COND
                                   ((NUMBERP CHANGEFLG)
                                    CHANGEFLG)
                                   (T                          ; Means change all occurrences.
                                     -1)))
                    (SETQ TOPLVL NIL)
                    (SETQ C3 (EDITFPAT1 C3))
                    [AND CHARFLG (NLISTP PAT)
                         (NLISTP C3)
                         [SETQ PAT (CONS 'Ÿ (CONS 'Ÿ (NCONC1 (UNPACK PAT)
                                                              'Ÿ]
                         (SETQ C3 (CONS 'Ÿ (CONS 'Ÿ (NCONC1 (UNPACK C3)
                                                             'Ÿ]
                    ;; If CHARFLG is T and neither pattern nor format contain alt-modes, supply them, i.e.  user wants a character replacement
                    ;; operation.  This option is used by the RC and RC1 commands, and by ESUBST.

                    )
             [(EQ C3 'N)
              (SETQ N 1)
              [COND
                 ((NLISTP (CAR L))
                  (GO LP1))
                 ((EQ (CAAR L)
                      CLISPTRANFLG)
                  (SETQ X (CADDAR L)))
                 (T (SETQ X (CAAR L]
              (SETQ LL (CONS X L))
              (COND
                 ((AND (NLISTP X)
                       UPFINDFLG)
                    ;; E.g. If at (COND --) and do F COND, cannot be allowed to match with this COND, as the subsequent UP would leave you
                    ;; right where you started.  However, if UPFINDFLG is NIL, then it is ok to match with this COND.
                  (GO LP1]
             (T (SETQ N C3)))
          (COND
             ((NOT (NUMBERP N))
              (SETQ N 1)))
          [COND
             ([COND
                 [(TAILP (CAR LL)
                         (CADR LL))
                  (AND (EQ (CAR (LISTP PAT))
                           '|...|)
                       (EDIT4E (CDR PAT)
                               (CAR LL]
                 (T (EDIT4E PAT (CAR LL]
              ;; This EDIT4E check is necessary because once search starts, EDIT4F1 is always looking down one level, i.e. at car's of list it is
              ;; examining.  Similarly, since once the search starts, tails are only matched against patterns beginning with ..., we do not call EDIT4E
              ;; here on a TAIL unless the pattern also begins with ...
              (COND
                 [CHANGEFLG (COND
                               ([NULL (AND (EQ PAT '&)
                                           (LISTP (CAR L]
                                ;; R can't work if you are already there, e.g. current expression is B and user says (R B C), or current
                                ;; expression is (CAR X) and user says (R (CAR X) (CDR Y)).  The AND check is to enable commands like
                                ;; (r1 & $.) to work.  In this case, it is assumed that & meant the first element in the current expression, not
                                ;; the current expression itself.
                                (PRIN1 "can't
                                       " T T)
                                (ERROR!]
                 ((ZEROP (SETQ N (SUB1 N)))
                  (RETURN (SETQ L LL]
          (SETQ X (CAR LL))
    LP    (COND
             [(EDIT4F1 PAT X MAXLEVEL TAIL)
              (AND (CDR L)
                   (SETQ UNFIND L))
              (RETURN (CAR (SETQ L (NCONC (CAR FF)
                                          (COND
                                             ((EQ (CADR FF)
                                                  (CAR LL))      ; To avoid repetitions.
                                              (CDR LL))
                                             (T LL]
             (TOPLVL (GO ERROR))
             ((EQ CHANGEFLG T)
              ;; R command only affects current expression.  However, R1 is equivalent to an F and then a replacement and so is allowed to search
              ;; above the current expression.
              (COND
                 (NEWFLG (RETURN T)))
              (GO ERROR)))
```

```
      LP1 (SETQ X (CAR LL))                                          ; Ascend from this element and begin searching the next element
                                                                     ; in the next higher list.
          (COND
             ((NULL (SETQ LL (CDR LL)))
              (COND
                 (NEWFLG                                             ; This was a replacement operation which has found a
                                                                     ; successful match.
                         (RETURN T)))
              (GO ERROR))
             ([SETQ TAIL (COND
                           ((AND (EQ X (CAR LASTAIL))
                                 (TAILP LASTAIL (CAR LL)))
                            ;; This is sort of an open UP.  It is necessary to handle the case where the current expression is atomic and the
                            ;; next higher expression contains two instances of it.
                            LASTAIL)
                           (T (MEMB X (CAR LL]
              (SETQ X (CDR TAIL))
              (GO LP)))
          (GO LP1)
      ERROR
          (SETQ COM PAT0)
          (ERROR!])
```

(**EDIT4F1**

```
  [LAMBDA (PAT X LVL TAIL)                                           (* wt%: " 5-APR-78 11:07")
```

;; In most cases, EDIT4F1 treats X as a list, and matches PAT against elements of X. However, if TAIL is not NIL, EDIT4F1 will also look at X itself ;; if (1) X is not a list (this covers the case where a list ends in an atom other than NIL), or (2) PAT begins with ... In both cases, X is EQ to CDR of ;; TAIL, and TAIL is used if replacement is being carried out.

```
      (PROG ((L L)
             TEM XX)
            (AND CHANGEFLG (NEQ X (CAR L))
                 (SETQ L (CONS X L)))
```

;; So that if there are any replacements in CLISP expressions that have been translated, editsmash will know to remove the translations.

```
            [COND
               ((AND (LISTP X)
                     (NULL TAIL)
                     (EQ (CAR X)
                         CLISPTRANFLG))
                (SETQ XX X)
                (SETQ TAIL (CDR X))
                (SETQ X (CDDR X]
       LP   (COND
               ((AND (LISTP PAT)
                     (EQ (CAR PAT)
                         '|...|))
```

;; This check is made before the NULL check because F (...) is acceptable and means find the first list ending in NIL.

```
                (GO CHECK...))
               ((NULL X))
               ((AND LVL (NOT (IGREATERP LVL 0)))                    ; NIL = infinity.
                (PRIN1 '"maxlevel exceeded.
                       " T))
               ((LISTP X)
                (GO ELEMENT))
               ((AND TAIL (SETQ TEM (EDIT4E PAT X CHANGEFLG)))       ; Compares PAT with atomic tail of a list.
                [COND
                   (CHANGEFLG (SETQ X (EDIT4F2 TAIL TEM C3 T]
                (COND
                   ((ZEROP (SETQ N (SUB1 N)))
                    (GO SUCC)))
```

;; Note that the current expression is left at the (atomic) tail to prevent accidents like (MOVE FOO TO ...) and FOO is CDR of (FIE . ;; FOO)

```
                ))
            (RETURN NIL)
       CHECK...
            (COND
               [(AND TAIL (SETQ TEM (EDIT4E (CDR PAT)
                                            X CHANGEFLG)))           ; Note that at this point, X may still be atomic, as in F (... . B)
                [COND
                   (CHANGEFLG (SETQ X (EDIT4F2 TAIL TEM C3 T]
                (COND
                   ((ZEROP (SETQ N (SUB1 N)))
                    (GO SUCC))
                   (CHANGEFLG
```

;; Don't want to go to LP1 because you don't want to search through new structure inserted by replacement.

```
                           (RETURN NIL))
                   ((NLISTP X)
                    (RETURN NIL))
                   (T (GO LP1]
               ((NLISTP X)
                (RETURN NIL))
```

```
            (T                                                    ; PAT is a ... pattern, so don't compare it with elements.
               (GO DESCEND)))
      ELEMENT
           [COND
              ((SETQ TEM (EDIT4E PAT (CAR X)
                                 CHANGEFLG))
               (COND
                  (CHANGEFLG (EDIT4F2 X TEM C3)))
               (COND
                  ((ZEROP (SETQ N (SUB1 N)))
                   [COND
                      ((OR (NULL UPFINDFLG)
                           (LISTP (CAR X)))
                       ;; Instead of adding atom and then doing UP --- this check is made and atom not added if UPFINDFLG is T.
                       (SETQ LASTAIL X)                           ; For use by UP.
                       (SETQ X (CAR X]
                   (GO SUCC))
                  (CHANGEFLG
                         ;; Don't want to go to DESCEND because you don't want to search through new structure inserted by replacement
                         ;; operation.
                         (GO LP1]
      DESCEND
           (COND
              ((AND (NULL TOPLVL)
                    (LISTP (CAR X))
                    (EDIT4F1 PAT (CAR X)
                             (AND LVL (SUB1 LVL)))
                    (ZEROP N))
               (SETQ X (CAR X)))
              (T (GO LP1)))
      SUCC
           (AND XX (EQ X (CDDR XX))
                (SETQ X XX))                                      ; CLISP expression.
           (COND
              ([AND FF (NOT (AND X (EQ X (CADR FF]                ; To eliminate repetitions.
               (TCONC FF X)))
           (RETURN (OR FF T))
      LP1  (SETQ TAIL X)
           (SETQ X (CDR X))
           (AND LVL (SETQ LVL (SUB1 LVL)))
           (GO LP])
```

(EDIT4F2

```
  [LAMBDA (NODE MATCH FORMAT CDRFLG)                              ; Edited 10-Nov-87 14:37 by jds
```

;; Analagous to CONSTRUCT in FLIP, with EDITFPAT1 playing the role of FORMTRAN. Replaces CAR of NODE by FORMAT (CDR if ;; CDRFLG=T). MATCH is the value returned by EDIT4E. If MATCH is a list of pointers and FORMAT begins with $, EDIT4F2 assembles a new ;; atom or string, replacing those sequences not matched by alt-modes with elements from NEW. For example, user types (R $1 $2) then all ;; terminal 1's will be changed to 2's.

```
    (PROG ([X (COND
                 (CDRFLG (CDR NODE))
                 (T (CAR NODE]
           FLG)
          (SETQ NEWFLG T)                                         ; to let EDIT4F know that a successful match was found.
          (SETQ FORMAT (EDIT4F3 FORMAT MATCH X))
          (COND
             ((EQ EDITQUIETFLG T)
              (GO OUT))
             ((NEQ MATCH T)                                       ; EDIT4E printed X.

              )
             (FLG
                   ;; MATCH was T, indicating no alt-modes, and therefore X was not printed by EDIT4E1. However, FLG being T means a format
                   ;; was used, and therefore X must be printed here. For example, (R FOO $1)
                   (PRIN2 X T T))
             (T (GO OUT)))
          (PRIN1 "->" T)
          (PRINT FORMAT T T)
      OUT [COND
             (CDRFLG (EDITSMASH NODE (CAR NODE)
                                FORMAT))
             (T (EDITSMASH NODE FORMAT (CDR NODE]
          (EDITSMASH1 FORMAT)
          (RETURN FORMAT])
```

(EDIT4F3

```
  [LAMBDA (FORMAT MATCH X)                                        (* lmm "18-NOV-82 13:54")
    (PROG (LST)
          (COND
             [(LISTP FORMAT)
              (COND
                 ([EQ (CAR FORMAT)
                      (CONSTANT (CHARACTER (CHARCODE ESCAPE]
```

```
                (SETQ FLG T))
               (T (RETURN (CONS (EDIT4F3 (CAR FORMAT)
                                        MATCH X)
                                (EDIT4F3 (CDR FORMAT)
                                        MATCH X]
          (T (RETURN FORMAT)))
      LP  [COND
             [(NLISTP (SETQ FORMAT (CDR FORMAT)))
              (RETURN (COND
                         ((AND (EQ MATCH T)
                               (NULL (CDR LST)))
                          (CAR LST))
                         ((STRINGP X)
                          (CONCATLIST LST))
                         (T (PACK LST]
             [[EQ (CAR FORMAT)
                  (CONSTANT (CHARACTER (CHARCODE ESCAPE]
              (SETQ LST (NCONC LST (COND
                                      ((EQ MATCH T)             ; Permits user to say (R FOO $1) meaning change all FOO's to
                                                                ; FOO1's, etc.
                                       (LIST X))
                                      (T (PROG1 (LDIFF (CAAR MATCH)
                                                       (CDAR MATCH))
                                                (SETQ MATCH (CDR MATCH)))]
             (T (SETQ LST (NCONC1 LST (CAR FORMAT]
          (GO LP])
```

(**EDITFPAT**

```
[LAMBDA (PAT FLG)                                               (* wt%: "23-NOV-76 1 45")
```

;; Done once at beginning of find operation. Replaces atoms ending in alt-modes with patterns recognized by EDIT4E. Analagous to PATTRAN in ;; FLIP, with role of MATCH being played by EDIT4E1.

```
(PROG (TEM)
      (RETURN (COND
                 [(LISTP PAT)
                  (COND
                     ((OR (EQ (CAR PAT)
                              '==)
                          (EQ (CAR PAT)
                              'Ÿ)
                          (EQ (CAR PAT)
                              'ŸŸ))
                      PAT)
                     (T (CONS (EDITFPAT (CAR PAT))
                              (EDITFPAT (CDR PAT]
                 ((OR (EQ PAT 'Ÿ)
                      (NOT (STRPOS 'Ÿ PAT)))
                  PAT)
                 [(STRPOS '"ŸŸ" PAT -2)                          ; Used to specify a search for a 'close' word using SKOR. See
                                                                ; comment in EDIT4E.
                  (SETQ TEM (CHCON PAT))
                  (FRPLACD (NLEFT TEM 3))
                  (CONS 'ŸŸ (CONS (LENGTH TEM)
                                 (CONS (PROG ((ND 0)
                                              CHAR)
                                             [MAPC TEM (FUNCTION (LAMBDA (X)
                                                                    (COND
                                                                       ((EQ X CHAR)
                                                                        (SETQ ND (ADD1 ND)))
                                                                       (T (SETQ CHAR X]
                                             (RETURN ND))
                                       TEM]
                 (T (CONS 'Ÿ (COND
                                (FLG (DUNPACK PAT CHCONLST1))
                                (T (UNPACK PAT])
```

(**EDITFPAT1**

```
[LAMBDA (X)                                                     (* rmk%: " 6-JUN-82 15:15")
```

;; Analgous to FORMTRAN in FLIP, with EDIT4F2 playing the role of CONSTRUCT. Used by EDIT4F once at the beginning of a find operation ;; that also specifies replacement --- i.e. an R command. Converts an atom or string containing alt modes into a list of the character sequences, ;; e.g. if X is ABCDEF$ then the value of EDITFPAT1 is ($ $ ABC $ DEF $) (The first $ is merely a flag.)

```
(COND
   ((OR (LITATOM X)
        (STRINGP X))
    (COND
       [(STRPOS 'Ÿ X)
        (CONS 'Ÿ (PROG((N 1)
                       (NC (NCHARS X))
                       VAL)
                 LP  (SETQ VAL (CONS [COND
                                       ((EQ (NTHCHARCODE X N)
                                            (CHARCODE ESCAPE))
                                        'Ÿ)
                                       (T (SUBSTRING X N (SETQ N (SUB1 (OR (STRPOS "Ÿ" X N)
```

```
                                                                    0]
                                        VAL))
                           [COND
                              ((OR (EQ N -1)
                                   (IGREATERP (SETQ N (ADD1 N))
                                              NC))
                               (RETURN (DREVERSE VAL]
                           (GO LP]
          (T X)))
      [(LISTP X)
       (CONS (EDITFPAT1 (CAR X))
             (EDITFPAT1 (CDR X]
      (T X])
```

(EDITFINDP

```
[LAMBDA (X PAT FLG)
```

(* Allows the user to use the edit find operation as a predicate without being inside the editor or doing any conses.)

```
  (PROG ((N 1)
         CHANGEFLG LASTAIL TOPLVL FF)
        (AND (NULL FLG)
             (SETQ PAT (EDITFPAT PAT T)))
        (RETURN (OR (EDIT4E PAT X)
                    (EDIT4F1 PAT X MAXLEVEL])
```

(FEDITFINDP

```
[LAMBDA (LST AT)                                        (* lmm "26-JUL-83 20:55")
  (OR (EQ AT LST)
      (AND (LISTP LST)
           (OR (FEDITFINDP (CAR LST)
                           AT)
               (FEDITFINDP (CDR LST)
                           AT])
```

(EDITBELOW

```
[LAMBDA (PLACE DEPTH)                                   ; See comment in EDITCOML
  (PROG ((L0 (PROG ((L L)
                    (LCFLG '_))
                   (EDITCOM PLACE)
                   (RETURN L)))
         L1 N)
        (COND
           ((NULL DEPTH)
            (SETQ COM C)
            (SETQ DEPTH 1))
           ((MINUSP (SETQ COM (EVAL DEPTH)))            ; If anything goes wrong from hhe on, the error message shuld
                                                        ; print the value of DEPTH.
            (ERROR!))
           (T (SETQ DEPTH COM)))
        (SETQ L1 (REVERSE L))
        (SETQ L0 (FMEMB (CAR L0)
                        L1))
    LP  [COND
           ((NULL L0)
            (ERROR!))
           [(ZEROP DEPTH)
            (FRPLACD L0)
            (SETQ UNFIND L)
            (RETURN (SETQ L (DREVERSE L1]
           ((NOT (TAILP (CADR L0)
                        (CAR L0)))
            (SETQ DEPTH (SUB1 DEPTH]
        (SETQ L0 (CDR L0))
        (GO LP])
```

(EDITBF

```
[LAMBDA (PAT N)
  (PROG ((LL L)
         X Y (FF (CONS)))
```

;; Same as EDIT4F, except searches in reverse printorder. If N is T (or at top level) search includes current expression, otherwise starts with first ;; expression that would be printed before the current expression.

```
        (SETQ COM PAT)
        (SETQ PAT (EDITFPAT PAT))
        (COND
           ((OR (NLISTP (CAR LL))
                (AND (NULL N)
                     (CDR LL)))                         ; Do not examine current expression.
            (GO LP1)))
    LP  [COND
           ((EDITBF1 PAT (CAR LL)
                     MAXLEVEL Y)
```

```
            (SETQ UNFIND L)
            (RETURN (CAR (SETQ L (NCONC (CAR FF)
                                        (COND
                                           ((EQ (CAR LL)
                                                (CADR FF))
                                            (CDR LL))
                                           (T LL]
     LP1 (SETQ X (CAR LL))
         (COND
            ((NULL (SETQ LL (CDR LL)))
             (ERROR!))
            ([OR (SETQ Y (MEMB X (CAR LL)))
                 (SETQ Y (TAILP X (CAR LL]
             (GO LP)))
         (GO LP1])
```

(EDITBF1

```
  [LAMBDA (PAT X LVL TAIL)
    (PROG [Y XX (...PAT (AND (LISTP PAT)
                             (EQ (CAR PAT)
                                 '|...|]
          (AND (LISTP X)
               (EQ (CAR X)
                   CLISPTRANFLG)
               (SETQ XX X)
               (SETQ X (CDDR X)))
     LP   [COND
            ((AND LVL (NOT (IGREATERP LVL 0)))
             (PRIN1 '"maxlevel exceeded.
                    " T)
             (RETURN NIL))
            ((EQ TAIL X)
             (RETURN (COND
                        ((AND (NOT ...PAT)
                              (EDIT4E PAT X))
                         ;; Only compare with X after you have searched inside it, e.g. if backing up to (COND -- (COND --)) should find inner
                         ;; COND.
                         (TCONC FF X]
          (SETQ Y X)
     LP1  (COND
            ([NULL (OR (EQ (CDR Y)
                           TAIL)
                       (NLISTP (CDR Y]                    ; TAIL is where you were last time.  Go until you find the tail
                                                          ; before it.
             (SETQ Y (CDR Y))
             (GO LP1)))
          (SETQ TAIL Y)                                   ; Y is a tail of X, TAIL is CDR of Y.
          (COND
            ((AND PAT (CDR TAIL)
                  (NLISTP (CDR TAIL))
                  (EDIT4E PAT (CDR TAIL)))                ; Atomic tail.
             (SETQ TAIL (CDR TAIL)))
            ((AND ...PAT (EDIT4E (CDR PAT)
                                 (CDR TAIL)))
             (SETQ TAIL (CDR TAIL)))
            ([AND (LISTP (CAR TAIL))
                  (EDITBF1 PAT (CAR TAIL)
                           (AND LVL (SUB1 LVL]            ; Descend first before comparing with outer one.
             (SETQ TAIL (CAR TAIL)))
            [(AND (NOT ...PAT)
                  (EDIT4E PAT (CAR TAIL)))
             (COND
                ((OR (NULL UPFINDFLG)
                     (LISTP (CAR TAIL)))
                 (SETQ LASTAIL TAIL)
                 (SETQ TAIL (CAR TAIL]
            (T (AND LVL (SETQ LVL (SUB1 LVL)))
               (GO LP)))
          (AND XX (EQ TAIL (CDDR XX))
               (SETQ TAIL XX))
          (COND
            ([NOT (AND TAIL (EQ TAIL (CADR FF]
             (TCONC FF TAIL)))
          (RETURN FF])
```

(EDITNTH

```
  [LAMBDA (X N)
```

;; If N is non-numeric, EDITELT is called, so that one can give commands such as (BI COND SETQ) meaning do a BI starting at the element
;; containing COND up to the one containing SETQ.

```
    (PROG (TEM)
          [COND
             ((NLISTP X)
              (ERROR!))
```

```
            ((EQ (CAR X)
                 CLISPTRANFLG)
             (SETQ X (CDDR X]
          (RETURN (COND
                     ((NOT (NUMBERP N))
                      ;; Normally EDITELT returns the element of this level list containing N.  However, if N is atomic and ends with an alt-mode,
                      ;; it will fail the first FMEMB, and EDITELT will return the tail of the list, so the second MEMB will fail.  This is the reason for
                      ;; the TAILP.
                      (OR (MEMB N X)
                          (MEMB (SETQ N (EDITELT N (LIST X)))
                                X)
                          (TAILP N X)))
                     ((ZEROP N)
                      (ERROR!))
                     ([SETQ TEM (COND
                                   ((MINUSP N)
                                    (NLEFT X (IMINUS N)))
                                   (T (NTH X N]
                      TEM)
                     (T (SETQ COM N)
                        (ERROR!])
```

(BPNT

```
  [LAMBDA (X)                                                              (* wt%: "14-MAY-76 18 42")
    (PROG (Y N Z)
          [COND
             ((ZEROP (CAR X))
              (SETQ Y (CAR L))
              (SETQ Z (CADR L)))
             (T (SETQ Y (CAR (EDITNTH (CAR L)
                                      (CAR X]
          [COND
             ((NULL (CDR X))
              (SETQ N 1))
             ([NULL (NUMBERP (SETQ N (CADR X]
              (ERROR!))
             ((MINUSP N)
              (SETQ N (ADD1 N)))
             (T                                                            ; Makes (P 0 N) have same effect as it did in old system.
                (SETQ N (SUB1 N]
          (RETURN (BPNT0 Y T N (OR (CADDR X)
                                   20)
                           Z])
```

(BPNT0

```
  [LAMBDA (X FILE CARLVL CDRLVL TAIL)                                      (* wt%: 11-MAY-76 18 0)
    (COND
       ((NULL (NLSETQ (LVLPRINT X FILE CARLVL CDRLVL TAIL)))
        (SETQ COM NIL)
        (ERROR!])
```

(EDIT.RI

```
  [LAMBDA (M N X)
    (PROG (A B)
          (SETQ A (EDITNTH X M))
          (SETQ B (EDITNTH (CAR A)
                           N))
          (COND
             ((OR (NULL A)
                  (NULL B))
              (ERROR!)))
          [PROG ((L (CONS (CAR A)
                          L)))
                ;; The only reason for this is so that EDITSMASH will also check (CAR a) for clisp translation.  Note that EDIT.RI is the only command
                ;; which lets you change something INSIDE of (CAR L) (The R command for xample is rebinding L as it goes down.)
                (MAPC (CDR B)
                      (FUNCTION EDITSMASH1))
                (EDITSMASH1 (CAR A))
                (EDITSMASH A (CAR A)
                           (EDITNCONC (CDR B)
                                      (CDR A]
          (EDITSMASH B (CAR B])
```

(EDIT.RO

```
  [LAMBDA (N X)
    (SETQ X (EDITNTH X N))
    (COND
       ((OR (NULL X)
            (NLISTP (CAR X)))
        (ERROR!)))
    (EDITSMASH (SETQ N (LAST (CAR X)))
```

```
          (CAR N)
          (CDR X))
    (EDITSMASH X (CAR X))
    (EDITSMASH1 (CAR X])
```

(EDIT.LI

```
  [LAMBDA (N X)
    (SETQ X (EDITNTH X N))
    (COND
       ((NULL X)
        (ERROR!)))
    (EDITSMASH X (CONS (CAR X)
                       (CDR X)))
    (EDITSMASH1 (CAR X))
    (EDITSMASH1 (CAR X])
```

(EDIT.LO

```
  [LAMBDA (N X)
    (SETQ X (EDITNTH X N))
    (COND
       ((OR (NULL X)
            (NLISTP (CAR X)))
        (ERROR!)))
    (EDITSMASH X (CAAR X)
          (CDAR X))
    (MAPC X (FUNCTION EDITSMASH1])
```

(EDIT.BI

```
  [LAMBDA (M N X)                                          (* lmm "26-JUL-83 20:51")
    (PROG (A B)
          (OR N (SETQ N M))
          [SETQ B (CDR (SETQ A (EDITNTH X N]
          (SETQ X (EDITNTH X M))
          (COND
             ((AND A (TAILP A X))
              (EDITSMASH A (CAR A))
              (EDITSMASH X (CONS (CAR X)
                                 (CDR X))
                     B)
              (EDITSMASH1 (CAR X)))
             (T (ERROR!])
```

(EDIT.BO

```
  [LAMBDA (N X)
    (SETQ X (EDITNTH X N))
    (COND
       ((NLISTP (CAR X))
        (ERROR!)))
    (EDITSMASH X (CAAR X)
           (EDITNCONC (CDAR X)
                  (CDR X)))
    (EDITSMASH1 (CAR X])

)

(RPAQ? EDITRDTBL (COPYREADTABLE "OLD-INTERLISP-T"))

(ADDTOVAR USERMACROS [ED NIL (E (ED (COND ((LISTP (%##))
                                           (CAR (%##)))
                                          (T (%##])

(ADDTOVAR EDITCOMSA ED)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: EDITBLOCK TTY/EDITL EDITL0 EDITL1 UNDOEDITL EDITCOM EDITCOMA EDITCOML EDITMAC EDITCOMS EDIT!UNDO
       UNDOEDITCOM UNDOEDITCOM1 EDITCOM1 EDITSMASH EDITSMASH1 EDITNCONC EDITAPPEND EDIT1F EDIT2F EDITNTH BPNT
       BPNT0 EDIT.RI EDIT.RO EDIT.LI EDIT.LO EDIT.BI EDIT.BO EDITDEFAULT EDITDEFAULT1 %## EDUP EDIT* EDOR EDRPT
       EDLOC EDLOCL EDIT%: EDITMBD EDITXTR EDITELT EDITCONT EDITSW EDITMV EDITTO EDITBELOW EDITRAN EDITSAVE
       EDITSAVE1 EDITH (ENTRIES TTY/EDITL EDITL0 %## UNDOEDITL BPNT0 EDITCONT EDLOCL)
       (SPECVARS L ATM COM LCFLG %#1 %#2 %#3 UNDOLST UNDOLST1 LASTAIL MARKLST UNFIND LASTP1 LASTP2 COMS
              EDITCHANGES EDITHIST0 LISPXID USERHANDLE)
       (RETFNS EDITL0 EDITL1)
       (BLKAPPLYFNS EDIT%: EDITMBD EDITMV EDITXTR EDITSW)
       (BLKLIBRARY NTH LAST MEMB NLEFT)
       (NOLINKFNS PRINTDEF EDITRACEFN EDITUSERFN)
       (LOCALFREEVARS FINDFLAG EDITHIST UNDOLST1 COM L L0 COM0 UNDOLST EDITLFLG ATM MARKLST EDITHIST0 UNFIND
              TYPEIN LCFLG LASTP1 LASTP2 LASTAIL COPYFLG ORIGFLG COMS TOFLG C LVL EDITCHANGES EDITLISPFLG)
       (GLOBALVARS EDITCALLS P.A.STATS EDITUNDOSTATS EDITUNDOSAVES SPELLSTATS1 P.A.STATS EDITUSERFN EDITIME
              DONTSAVEHISTORYCOMS COMPACTHISTORYCOMS EDITEVALSTATS MAXLOOP EDITCOMSL EDITCOMSA DWIMFLG
              CLISPTRANFLG EDITOPS HISTORYCOMS REREADFLG HISTSTR3 EDITRDTBL EDITHISTORY HISTSTR0 LISPXHISTORY
              LISPXBUFS EDITRACEFN EDITMACROS USERMACROS CLISPARRAY CHANGESARRAY COMMENTFLG **COMMENT**FLG
              EDITSMASHUSERFN))
```

```
(BLOCK%: EDITFINDBLOCK EDIT4E EDIT4E1 EDITQF EDIT4F EDITFPAT EDITFPAT1 EDIT4F1 EDIT4F2 EDIT4F3 EDITSMASH
       EDITSMASH1 EDITFINDP EDITBF EDITBF1 ESUBST (ENTRIES EDIT4E EDIT4E1 EDITQF EDIT4F EDITFPAT EDITFINDP
                                                          EDITBF ESUBST)
       (LOCALFREEVARS C3 CHANGEFLG N TOPLVL FF NEWFLG FLG)
       (GLOBALVARS EDITUNDOSAVES CHCONLST2 EDITQUIETFLG CHCONLST1 MAXLEVEL UPFINDFLG CLISPTRANFLG CHANGESARRAY
              CLISPARRAY EDITHISTORY)
       (SPECVARS ATM L COM UNFIND LASTAIL UNDOLST1 EDITCHANGES))

(BLOCK%: NIL EDITFA TTY/EDITE (SPECVARS EDITCHANGES EDITFN))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS FILELST DWIMFLG DWIMWAIT DWIMLOADFNSFLG)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA %##)

(ADDTOVAR NLAML EDITF2)

(ADDTOVAR LAMA )
)

(PUTPROPS EDIT COPYRIGHT ("Venue & Xerox Corporation" T 1983 1984 1985 1986 1987 1990))
```

FUNCTION INDEX

VARIABLE INDEX