File created: 29-Aug-91 13:42:59 **{PELE:MV:ENVOS}<LISPCORE>LIBRARY>DEDIT.;4**

changes to: (FNS SETUPDEDITW)
(RECORDS STACK)
(MACROS EDITBLOCKCALL CONTROLCODE OVERLAP SHIFTSELECTKEYS)

previous date: 27-Aug-91 13:00:13 {PELE:MV:ENVOS}<LISPCORE>LIBRARY>DEDIT.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS


```
(RPAQQ DEDITCOMS
       [(COMS                                                    ; User entry to the editor
              (PROP (Definition-for-EDITL Definition-for-EDITE Definition-for-EDITDATE)
                    DEDIT)
              [DECLARE%: DONTEVAL@LOAD DOCOPY (VARS (*DISPLAY-EDITOR* 'DEDIT]
              (FNS DEDITIT))
        (COMS                                                    ; Hooks between tty editor and DEDIT.  We redefine EDITL to
                                                                 ; get into DEDIT from system editor calls
              (FNS DEDITL DEDITL0 DEDITTTYFN))
        (COMS                                                    ; Basic DEDIT menu commands
              (FNS DEDITAfter DEDITBefore DEDITDelete DEDITReplace DEDITSwitch DEDITBI DEDITBO DEDITLI DEDITLO
                   DEDITRI DEDITRO DEDITUndo UNDOCHOOSE DEDITFind DEDITSwap DEDITCenter DEDITCopy DEDITReprint
                   DEDITEditCom DEDITARGS DEDITBreak DEDITEval DEDITExit)
              (FNS DEDITEdit DEDITCEdit DEDIT.TTYinEdit DEDITDatatype)
              (ADDVARS (DT.EDITMACROS)))
        (COMS                                                    ; Structure changing
              (FNS SETPTRTO DEDITCONS DEDITZAPCAR DEDITZAPCDR DEDITZAPNODE DEDITZAPBOTH DEDITFZAP DEDITZAPCLISP
                   DEDITZAPCHANGES DEDITMOVETAILDOWN DUNDOEDITL DUNDOEDITCOM DUNDOEDITCOM1))
        (COMS                                                    ; Selection code.  Select expressions or from the command
                                                                 ; menu
              (FNS DEDITSLCTLP DEDITUSER SELECTKEYS DODEDITTYPEDCOM DEDITREADLINE SHADEIFNOTBUF DEDITBUTTONFN
                   DEDITRIGHTBUTTONFN DEDITWINDOWENTRYFN SELECTELEMENT SELECTREAD SELECTTREE SEARCHMAP WITHINME
                   ONAPARENP SELECTDONE INWINDOW FINDLCA DOMINATE?)
              (ALISTS (DEDITTYPEINCOMS F S Z))
              (PROP VARTYPE DEDITTYPEINCOMS))
        (COMS                                                    ; Handling the selection stack
              (FNS POPSELECTION PUSHSELECTION NXTSELECTION TOPSELECTION SWITCHANDSHADE SHADESELECTION
                   SHADESELECTION1 SHADESELECTION2 OVERLAPSELBAND PUSHEDITCHAIN MAKESELCHAIN PUSHINTOBUF
                   DUMMYMAPENTRY FLIPSELS FLIPSELSIN FIXUPSEL NEWSELFOR))
        (COMS                                                    ; Initializing and flushing edit windows
              (FNS ACTIVEEDITW FINDEDITW GETEDITW GETDEDITDEF4 MAKEEDITW NAMEOFEDITW PURGEW MAKECPOSBE SAMEEDITW
                   SETUPDEDITW TOPEDITW UNDEDITW WHICHEDITW ZORCHEDITW ZORCHEDWP UNZORCHME)
              (INITVARS (DEditLinger T)))
        (COMS                                                    ; Manipulating the Edit menu
              (FNS SETEDITMENU CACHEDEDITCOMS FINDEDITCOM READEDITMENU SHADEMENUENTRY DEDITMENURESTORE)
              [VARS (*DEDIT-MENU-COMMANDS* '((After DEDITAfter)
                                             (Before DEDITBefore)
                                             (Delete DEDITDelete)
                                             (Replace DEDITReplace)
                                             (Switch DEDITSwitch)
                                             ("( )" DEDITBI ("( ) in" DEDITBI)
                                                    ("( in" DEDITLI)
                                                    (") in" DEDITRI))
                                             ("( ) out" DEDITBO ("( ) out" DEDITBO)
                                                    ("( out" DEDITLO)
                                                    (") out" DEDITRO))
                                             (Undo DEDITUndo (Undo DEDITUndo)
                                                   (!Undo (DEDITUndo T))
                                                   (?Undo (UNDOCHOOSE))
                                                   (&Undo (UNDOCHOOSE T)))
                                             (Find DEDITFind)
                                             (Swap DEDITSwap (Center DEDITCenter)
                                                   (Clear (SETQ \DEDITSELECTIONS NIL))
                                                   (Copy DEDITCopy)
                                                   (Pop (POPSELECTION))
                                                   (Swap DEDITSwap))
                                             (Reprint DEDITReprint)
                                             [Edit DEDITEdit [DEdit (DEDITEdit 'DISPLAY 'Def)
                                                                    NIL
                                                                    (SUBITEMS ("DEdit Def" (DEDITEdit
                                                                                             'DISPLAY
                                                                                             'Def))
                                                                              ("DEdit Form" (DEDITEdit 'DISPLAY
                                                                                                       'Form]
                                                   [TTYEdit (DEDITEdit 'TELETYPE 'Def)
```

```
                                                    NIL
                                                    (SUBITEMS ("TTYEdit Def" (DEDITEdit 'TELETYPE
                                                                                        'Def))
                                                           ("TTYEdit Form" (DEDITEdit 'TELETYPE
                                                                                      'Form]
                                                 (TTYIn% Form (DEDITEdit 'DEDIT.TTYinEdit 'Form]
                                           [EditCom DEDITEditCom (?= DEDITARGS)
                                                  (GETD (DEDITEditCom 'GETD))
                                                  (CL (DEDITEditCom 'CL))
                                                  (DW (DEDITEditCom 'DW))
                                                  (REPACK (DEDITEditCom 'REPACK))
                                                  (CAP (DEDITEditCom 'CAP))
                                                  (LOWER (DEDITEditCom 'LOWER))
                                                  (RAISE (DEDITEditCom 'RAISE]
                                           (Break DEDITBreak)
                                           (Eval DEDITEval)
                                           (Exit DEDITExit (OK DEDITExit)
                                                 (STOP (DEDITExit T]
      (GLOBALVARS *DEDIT-MENU-COMMANDS*))
(COMS                                                    ; Maintaining deditmap entries and the edit chain
      (FNS BUFSELP EDITWINDOWP GETLEFT GETMEBP HASASBP TAILOF DOTTEDEND GETME4 GETSELMAP DEARME DPCDRSEL
           GETDPME GETEBUF GETEBUFREGION GETEDITCHAIN GETDEDITMAP GETMAP? UNPURGEDP SUBSELOF SETDEDITMAP
           TAKEDOWN)
      (INITVARS (*DEDIT-BUFFER-HEIGHT* 60))
      (GLOBALVARS *DEDIT-BUFFER-HEIGHT*))
(COMS (FNS DEDITRESHAPEFN DEDITREPAINTFN)
      (FNS RESETDEDIT DEDITDATE DEDITMARKASCHANGED)
      (FNS COPYCONS COPYOUTCONS MAPENTRYP THELIST)
      (FNS CANT))
(DECLARE%: DOEVAL@COMPILE DONTCOPY (RECORDS STACK)
       (MACROS EDITBLOCKCALL CONTROLCODE OVERLAP SHIFTSELECTKEYS)
       (CONSTANTS (LINETHICKNESS 2)
              (PRIMSHADE 65535)
              (SECSHADE 3598)
              (SWITCHSHADE (LOGXOR PRIMSHADE SECSHADE))
              (READSHADE 23130)
              (CHANGEDSHADE 8840))
       (GLOBALVARS DEditWindow \DEDITMNUW \DEDITBUFW \DEDITALLOWSELS \DEDITWINDOWS \DEDITSELECTIONS
              DT.EDITMACROS UPFINDFLG)
       (SPECVARS ATM EDITCHANGES EDITHIST LASTAIL UNDOLST UNDOLST1))
(DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
                                        DEDITPP))
[DECLARE%: DONTEVAL@LOAD DOCOPY (FILES DSPRINTDEF NEWPRINTDEF DEDITPP)
       (P (AND (GETD 'RESETDEDIT)
               (RESETDEDIT]
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                       (NLAML)
                                                                       (LAMA CANT])
```

;; User entry to the editor

```
(PUTPROPS DEDIT Definition-for-EDITL DEDITL)

(PUTPROPS DEDIT Definition-for-EDITE TTY/EDITE)

(PUTPROPS DEDIT Definition-for-EDITDATE DEDITDATE)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(RPAQQ *DISPLAY-EDITOR* DEDIT)
)

(DEFINEQ
```

(**DEDITIT**

```
  [LAMBDA (EFN EARGS EMODE)                                    (* bas%: "21-MAR-83 20:38")
    (RESETFORM (EDITMODE EMODE)
           (APPLY EFN EARGS])

)
```

;; Hooks between tty editor and DEDIT. We redefine EDITL to get into DEDIT from system editor calls

```
(DEFINEQ
```

(**DEDITL**

```
  [LAMBDA (L COMS ATM MESS EDITCHANGES)                        (* bas%: "19-JUN-83 23:58")
```

(* Value is edit push-down list L. EDITCHANGES is used for destructively marking whether the edit made any changes.)

```
    (RESETLST
        (RESETSAVE \DEDITSELECTIONS (create STACK))
        (COND
           (COMS (RESETSAVE EDITMACROS (CONS '(TTY%: NIL (E (DEDITTTYFN ATM TYPE)
                                                            T))
                                             EDITMACROS))
```

```
            (NORMAL/EDITL L COMS ATM MESS EDITCHANGES))
      (T (AND MESS (printout PROMPTWINDOW .TAB0 0 MESS T))
         [PROG [MARKLST UNDOLST UNDOLST0 UNDOLST1 UNFIND LASTAIL TMP (EXPR (CAR (LAST L]
```

(* EXPR is the top level expression. 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 directly.)

```
          [COND
             ([OR (EQ EXPR (GETPROP 'EDIT 'LASTVALUE))
                  [AND ATM (EQ EXPR (SETQ TMP (GETPROP ATM 'EDIT-SAVE]
                  (SOME (CAR LISPXHISTORY)
                        (FUNCTION (LAMBDA (X)
                                  (EQ EXPR (SETQ TMP (CADR (MEMB '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.)

```
              (SETQ MARKLST (CADR TMP))
              (SETQ UNDOLST (CADDR TMP))
              (COND
                 ((CAR UNDOLST)                          (* Don't want to block it twice.)
                  (push UNDOLST NIL)))
              (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 TMP]
          (COND
             [(PROG1 (DEDITL0 EXPR (GETEDITW ATM (AND (BOUNDP 'TYPE)
                                                      TYPE)))
                                                      (* Even if some error occurs, still want to move undo information
                  to LISPXHISTORY.)
                  [COND
                     (UNDOLST1 (push UNDOLST (CONS T (CONS \DEDITSELECTIONS UNDOLST1]
                  (AND LISPXHIST (NEQ UNDOLST UNDOLST0)
                       (UNDOSAVE (LIST 'DUNDOEDITL \DEDITSELECTIONS UNDOLST UNDOLST0)
                             LISPXHIST))              (* Makes entire DEDITL undoable.)
                  )                                   (* Normal OK exit)
              (AND ATM (LITATOM ATM)
                   (REMPROP ATM 'EDIT-SAVE))
              [SETQ TMP (CONS EXPR (CONS MARKLST (CONS UNDOLST (LIST EXPR]
              (PUTPROP 'EDIT 'LASTVALUE TMP)
              (COND
                 (LISPXHIST (NCONC LISPXHIST (LIST 'EDIT TMP]
              (T (ERROR!]
        L)))])
```

(DEDITL0

```
  [LAMBDA (EXPR EDS SEL)                                  (* bvm%: "31-Jul-86 14:35")
```

(* * "DEDITL0 should only be called while under DEDITL or DEDITTTYFN since the global states of the edit are all bound there. Note that individual calls to DEDITL0 are not undoable, because structure changes are saved on UNDOLST1 and only moved to UNDOLST at the end of each command. DEDITL finally moves UNDOLST to LISPXHISTORY.")

```
    (RESETSAVE NIL (LIST 'SETCURSOR (CURSOR WAITINGCURSOR)))
    [LET ((PM (GETMAP? EDS))
          (ENV (DEDIT-MAKE-READER-ENV EXPR))
          OLDENV)
         (COND
            ((AND PM (EQ EXPR (fetch SELEXP of PM))
                  (SETQ OLDENV (WINDOWPROP EDS 'READER-ENVIRONMENT))
                  (EQUAL-READER-ENVIRONMENT OLDENV ENV))  (* "Editing the same thing that's in the window now, and in the
                                                          same reader environment")
             (TOTOPW EDS)                                 (* "Window might have been closed")
             )
            (T (WINDOWPROP EDS 'READER-ENVIRONMENT ENV)
               (SETUPDEDITW EDS (LIST EXPR]
    (AND SEL (PUSHEDITCHAIN SEL))                         (* "ERSETQ prevents UNDOLST lossage due to ^E")
    (ERSETQ (bind EDITHIST COM ACT SS do (until (SETQ COM (DEDITSLCTLP EDS)))
                                         (SETQ SS \DEDITSELECTIONS)
                                                          (* "Save selection stack")
                                         (SETQ ACT (CDR COM))
                                                          (* "Unpack CONS from READEDITMENU")
                                         (SETQ COM (CAR COM))
                                         [COND
                                            (EDITHISTORY (COND
                                                            ((PROG1 (AND ATM (NOT EDITHIST))
                                                                 (* First time thru)
                                                                 (EDITBLOCKCALL EDITSAVE COM)
                                                                 (* Sets EDITHIST)
                                                                 )
                                                             (LISPXPUT '*FIRSTPRINT* (LIST 'EDITL2 ATM T)
                                                                    NIL EDITHIST]
                                         (SETQ UNDOLST1 NIL)
                                                          (* "Holds any changes from execution of this command.")
                                         (COND
                                            [(PROG1 [ERSETQ (COND
                                                               ((LITATOM ACT)
```

```
                                                        (APPLY* ACT))
                                                       (T (EVAL ACT]
                                                 [COND
                                                    (UNDOLST1 (REPPCHANGES UNDOLST1)
                                                           (push UNDOLST (SETQ UNDOLST1 (CONS COM
                                                                                              (CONS SS UNDOLST1]
                                                 (COND
                                                    (EDITHIST    (* "Set in EDITSAVE.")
                                                           (RPLACA EDITHIST UNDOLST1)))))]
                                              (T                 (* "Restore selections")
                                                 (SETQ \DEDITSELECTIONS SS)))
                                                       (* "Only way out is a RETFROM via one of the exit fns")
                                           ])
```

(DEDITTTYFN

```
  [LAMBDA (NAME TYPE)                                   (* bas%: " 7-AUG-83 16:38")
                                                        (* Provides DEDIT interface to TTY%: commands from under
                                                        standard editor)
    (DECLARE (USEDFREE L LASTAIL))                      (* From EDITL0)
    (PROG [UNDOLST TEM (TE (CAR (LAST L]
          (RESETLST

          (* The RESETLST is for DEDITL0; the binding of UNDOLST1 protects the containing EDIT;
          TEM=T unless DEDITL0 was STOPed)

              (PROG (UNDOLST1)
                    (SETQ TEM (DEDITL0 TE (GETEDITW NAME TYPE)
                                       L))))
          (AND UNDOLST (push UNDOLST1 (CONS 'GROUPED UNDOLST)))
          (COND
             (TEM [SETQ L (OR (AND (SUBSELOF TE (TOPSELECTION T))
                                   (GETEDITCHAIN (TOPSELECTION T)))
                              (for I on L thereis (AND (SUBSELOF TE (CAR I))
                                                       (SETQ LASTAIL (CAR I]

          (* Reset edit chain only if current selection still points to some part of the expression being edited)

                )
             ([EVALV 'COMS (SETQ TEM (STKPOS 'EDITL0]
              (RETEVAL TEM '(ERROR!)
                       T))
             (T (SHOULDNT])

)

;; Basic DEDIT menu commands

(DEFINEQ

(DEDITAfter
  [LAMBDA NIL                                           (* bas%: "17-MAR-83 22:15")
    (PROG ([NU (COPY (CAR (POPSELECTION]
           (TGT (POPSELECTION)))
          (DEDITZAPCDR TGT (PUSHSELECTION (COND
                                             ((DPCDRSEL TGT)
                                              (DEDITCONS (CDR TGT)
                                                         NU TGT))
                                             (T (DEDITCONS NU (CDR TGT)
                                                           TGT])

(DEDITBefore
  [LAMBDA NIL                                           (* bas%: "16-MAR-83 12:40")
    (PROG ((SRC (POPSELECTION))
           (TGT (POPSELECTION)))
          (PUSHSELECTION (SETPTRTO TGT (DEDITCONS (COPY (CAR SRC))
                                                  (COND
                                                     ((DPCDRSEL TGT)
                                                      (CDR TGT))
                                                     (T TGT))
                                                  TGT])

(DEDITDelete
  [LAMBDA NIL                                           (* bas%: "16-MAR-83 11:51")
                                                        (* Deletes top elt from structure. Pushes it back on into the
                                                        buffer)
    (PROG ((S (POPSELECTION)))
          [PUSHINTOBUF (LIST (COPY (CAR S]

          (* Copy keeps structure in buffer separate from that on undolst, which may later get inserted back)

          (SETPTRTO S (COND
                         ((DPCDRSEL S)
                          NIL)
                         (T (CDR S])
```

```
(DEDITReplace
  [LAMBDA NIL                                                    (* bas%: " 5-JUL-83 23:50")
    (PROG ((SRC (POPSELECTION))
           (TGT (TOPSELECTION)))
          (DEDITZAPCAR TGT (SUBST (CAR TGT)
                                  (OR EDITEMBEDTOKEN (CONSTANT (PACK NIL)))
                                  (CAR SRC])

(DEDITSwitch
  [LAMBDA NIL                                                    (* bas%: "16-MAR-83 21:05")
    (PROG ((A (TOPSELECTION))
           (B (NXTSELECTION)))
          (COND
             ((OR (DOMINATE? A B)
                  (DOMINATE? B A))
              (CANT "Switch into oneself")))
          (DEDITZAPCAR A (PROG1 (CAR B)
                                (DEDITZAPCAR B (CAR A)))])

(DEDITBI
  [LAMBDA NIL                                                    (* bas%: "16-MAR-83 11:51")
    (PROG ((A (POPSELECTION))
           (B (POPSELECTION))
           C)
          (COND
             ((TAILOF B A))
             [(TAILOF A B)
              (SETQ A (PROG1 B (SETQ B A]
             (T (CANT "Not brothers!")))
          (COND
             ((DPCDRSEL B))
             (T (SETQ C (CDR B))                                 (* Done in this order in case A=B)
                (DEDITZAPCDR B NIL)))
          (DEDITZAPBOTH A (COPYCONS A)
                C)
          (PUSHSELECTION A])

(DEDITBO
  [LAMBDA NIL                                                    (* bas%: "12-Sep-84 14:37")
    (PROG ((TGT (POPSELECTION)))
          (DEDITMOVETAILDOWN TGT NIL)
          (SETPTRTO TGT (CAR TGT])

(DEDITLI
  [LAMBDA NIL                                                    (* bas%: " 2-MAR-83 11:33")
    (PROG ((A (TOPSELECTION)))
          (DEDITZAPBOTH A (COPYCONS A])

(DEDITLO
  [LAMBDA NIL                                                    (* bas%: " 2-MAR-83 11:34")
    (PROG ((A (TOPSELECTION)))
          (DEDITZAPNODE A (THELIST (CAR A])

(DEDITRI
  [LAMBDA NIL                                                    (* bas%: "30-Sep-84 13:19")
    (PROG (B (A (POPSELECTION)))
          (OR (CDR A)
              (CANT "RI at end of tail has no effect"))          (* Has no effect and scrambles undo list)
          [SETQ B (fetch TAIL of (GETMEBP (GETME4 A T]
          (DEDITMOVETAILDOWN B (CDR A))
          (DEDITZAPCDR A NIL)
          (PUSHSELECTION B])

(DEDITRO
  [LAMBDA NIL                                                    (* bas%: "12-Sep-84 14:40")
    (DEDITMOVETAILDOWN (TOPSELECTION)
           NIL])

(DEDITUndo
  [LAMBDA (END)                                                  (* bas%: "12-Sep-84 23:54")
    (bind FLG for LST on UNDOLST do (OR FLG (SETQ FLG (CAAR LST)))
                                    (DUNDOEDITCOM (CAR LST)
                                           T)
       repeatuntil (OR (NULL END)
                       (EQ END (CAR LST))
                       (NULL (CAR LST)))
       finally (OR FLG (CANT (COND
```

```
                              ((CDR LST)
                               "Undo blocked")
                              (T "Nothing saved"])
```

(UNDOCHOOSE

```
  [LAMBDA (THRUP)                                                    (* bas%: "22-Mar-84 23:14")
    (PROG [(C (RESETFORM (CURSOR DEFAULTCURSOR)
                      (MENU (create MENU
                                  ITEMS _ (APPEND (for I in UNDOLST
                                                     collect (LIST (OR (CAR I)
                                                                       (PACK* "* " (CADR I)
                                                                              " *"))
                                                                   (KWOTE I)))
                                                  (LIST (LIST '**TOP** NIL)))
                                  TITLE _ (COND
                                              (THRUP "Undo Thru")
                                              (T "Undo One"))
                                  CENTERFLG _ T]
          (COND
             ((NOT C))
             (THRUP (DEDITUndo C))
             (T (DUNDOEDITCOM C T])
```

(DEDITFind

```
  [LAMBDA NIL                                                        (* bas%: " 5-Apr-84 23:21")
    (PROG (LASTAIL L TGT UNFIND (COM 'Find))
          (DECLARE (SPECVARS L UNFIND COM))
          (SETQ L (GETEDITCHAIN (POPSELECTION)))                     (* Sets LASTAIL)
          (SETQ TGT (CAR (TOPSELECTION)))
          (COND
             ([ERSETQ (RESETVARS (UPFINDFLG)
                                 (EDIT4F TGT 'N]
              (PUSHEDITCHAIN L)                                      (* Uses LASTAIL)
              )
             (T (CANT TGT "Not found"])
```

(DEDITSwap

```
  [LAMBDA NIL                                                        (* bas%: "24-MAR-83 15:57")
    (replace TOPELT of \DEDITSELECTIONS with (PROG1 (NXTSELECTION)
                                                    (replace NXTELT of \DEDITSELECTIONS with (TOPSELECTION)))])
```

(DEDITCenter

```
  [LAMBDA (NOTIFVIS)                                                 (* bas%: "26-Mar-84 15:17")
    (PROG [AW WO WH (A (GETME4 (TOPSELECTION]
          (OR A (RETURN))
          (SETQ AW (WFROMDS (fetch PDSP of A)))
          (SETQ WO (WYOFFSET NIL AW))
          (SETQ WH (WINDOWPROP AW 'HEIGHT))
          (AND NOTIFVIS (OVERLAPSELBAND A (IPLUS WO WH)
                                WO)
               (RETURN))                                             (* Make sure the sel highlite is visible)
          (RESETVARS (\DEDITSELECTIONS)

          (* Supress selections as they are not up and the scrollw will otherwise take them down)

                    (SCROLLW AW 0 (IDIFFERENCE (IPLUS WO (IQUOTIENT (IDIFFERENCE WH
                                                                      (IDIFFERENCE (fetch STARTY
                                                                                      of A)
                                                                                   (fetch STOPY of A)))
                                                                    2))
                                               (fetch STOPY of A])
```

(DEDITCopy

```
  [LAMBDA NIL                                                        (* bas%: " 2-MAR-83 11:37")
    (PUSHINTOBUF (LIST (COPY (CAR (TOPSELECTION])
```

(DEDITReprint

```
  [LAMBDA NIL                                                        (* bas%: " 2-MAR-83 11:37")
    (REPP (GETME4 (TOPSELECTION)
                  T])
```

(DEDITEditCom

```
  [LAMBDA (C)                                                        (* bas%: "30-MAR-83 20:55")
    [OR C (SETQ C (CAR (POPSELECTION]
    (PROG (TYPE ATM EDITCHANGES LASTAIL LASTP1 LASTP2 TSM SCR (TS (POPSELECTION)))
          (DECLARE (SPECVARS TYPE ATM EDITCHANGES LASTAIL LASTP1 LASTP2))
                                                                     (* For DEDITL and EDITL0)
          [COND
             ((SETQ TSM (GETME4 TS))
              [COND
```

```
                ((SETQ SCR (WINDOWPROP (fetch PDSP of TSM)
                                       'DEDITWHOAMI))
                 (SETQ ATM (CAR SCR))
                 (SETQ TYPE (CADR SCR]
              (SETQ EDITCHANGES (WINDOWPROP (fetch PDSP of TSM)
                                            'DEDITCHANGES]
          (PUSHEDITCHAIN (EDITL0 (GETEDITCHAIN TS)
                                 (MKLIST C])
```

(DEDITARGS

```
  [LAMBDA (F)                                                    (* bas%: "26-Mar-84 15:18")
    (SETQ F (OR F (TOPSELECTION)))
    (while (LISTP F) do (SETQ F (CAR F)))
    (PUSHINTOBUF (LIST (CONS F (COPY (CAR (OR (AND (LITATOM F)
                                                   (NLSETQ (SMARTARGLIST F T)))
                                              '((not a function])
```

(DEDITBreak

```
  [LAMBDA NIL                                                    (* lmm " 1-JUL-84 23:33")
    (PROG (WHO AMP CARFORM (A (POPSELECTION)))
          (SETQ AMP (GETME4 A))
          [SETQ WHO (AND AMP (WINDOWPROP (fetch PDSP of AMP)
                                         'DEDITWHOAMI]
          (DEDITZAPCAR A (LIST 'BREAK1 (CAR A)
                               T
                               [BREAKINCOMMENT WHO (LIST 'AROUND (COND
                                                                    ((NLISTP (CAR A))
                                                                     (CAR A))
                                                                    (T (CAAR A]
                               NIL))
          (OR [COND
                 (AMP (AND (fetch BP of AMP)
                           (FMEMB (CAAR (fetch TAIL of (fetch BP of AMP)))
                                  NOBREAKS)
                           (PROMPTPRINT "Break installed inside a NOBREAKS"))
                      (COND
                         ((EQ (CADR WHO)
                              'FNS)
                          (/PUTPROP (CAR WHO)
                                    'BROKEN-IN T)
                          (/PUTPROP (CAR WHO)
                                    'BRKINFO
                                    (LIST (LIST (LIST 'AROUND CARFORM)
                                                NIL NIL)))
                          (/SET 'BROKENFNS (CONS (CAR WHO)
                                                 BROKENFNS]
              (PROMPTPRINT "Break installed, but not recorded"])
```

(DEDITEval

```
  [LAMBDA NIL                                                    (* bas%: "19-Mar-84 09:44")
    (PROG [(S (CAR (POPSELECTION)))
           (SP (STKNTH 2 'DEDITL0]                               (* There are various entry points.
                                                                 They all call DEDITL0 after having done an ERRORSET.)
          [PUSHINTOBUF (COND
                          ((LITATOM S)
                           (LIST (EVALV S SP)))
                          ((ERSETQ (ENVAPPLY (FUNCTION LISPXEVAL)
                                             (LIST S NIL)
                                             SP)))
                          (T (LIST 'NOBIND]
          (RELSTK SP])
```

(DEDITExit

```
  [LAMBDA (STOPFLG)                                              (* mjs "26-Mar-86 12:33")
    (AND EDITHIST ATM (NOT STOPFLG)
         (LISPXPUT '*PRINT* (LIST 'EDITL2 ATM)
                NIL EDITHIST))                                   (* Hoaky stuff for the edit history list)
    (RETFROM (FUNCTION DEDITL0)
           (NOT STOPFLG)
           T])

)

(DEFINEQ
```

(DEDITEdit

```
  [LAMBDA (EDITOR EDITEE)                                        (* bvm%: "30-May-86 16:50")
    (RESETLST
        (RESETSAVE (SETCURSOR DEFAULTCURSOR)
                   (LIST 'SETCURSOR WAITINGCURSOR))
        (PROG ((S (CAR (TOPSELECTION)))
               A)
              (SELECTQ EDITEE
```

```
                ((Def NIL)
                     (COND
                        ((NOT (OR (LISTP S)
                                  (LITATOM S)))
                         (DEDITDatatype S))
                        ((AND (for old (S _ (POPSELECTION)) by (CAR S) while (LISTP S)
                                  finally (RETURN (LITATOM S)))
                              (SETQ A (TYPESOF S NIL NIL '?))
                              (SETQ A (SELECT.ATOM.ASPECT S NIL A)))
                         (RESETSAVE (EDITMODE EDITOR))        (* User can refuse all SELECT.ATOM.ASPECT choices)
                         (EDITDEF S A '?))
                        (T (CANT "No editable aspect"))))
                (Form (AND [SETQ S (APPLY* (SELECTQ EDITOR
                                                    ((TELETYPE DISPLAY)
                                                         (RESETSAVE (EDITMODE EDITOR))
                                                         (FUNCTION EDITE))
                                                    EDITOR)
                                           (LIST (COPY S]
                           (DEDITZAPCAR (TOPSELECTION)
                                  (CAR S))))
                (SHOULDNT)))])
```

(DEDITCEdit

```
  [LAMBDA (E FN)                                              (* bvm%: "30-May-86 16:55")

          (* * "Edits an expression using the editor defined by FN.  FN takes 2 args, the first a list of the expression(s) to edit, the
          second the edit window.  Returns new list of expressions.")

    (LET ((EW (GETEBUF (TOPEDITW)))
          V)
         (SETQ V (APPLY* FN E EW))
         (COND
            ((CDR V)                                          (* Replaced one expression with many)
             (SETQ V (LIST V)))
            (T V))
         (OR (BUFSELP (GETME4 (TOPSELECTION)))
             (BUFSELP (GETME4 (NXTSELECTION T)))
             (SETUPDEDITW EW (COPY V)))
        V])
```

(DEDIT.TTYinEdit

```
  [LAMBDA (S)                                                 (* bvm%: "30-May-86 16:55")
    (COND
       ((DEFINEDP 'TTYINEDIT)
        (DEDITCEdit S 'TTYINEDIT))
       (T (CANT "TTYIN not loaded"])
```

(DEDITDatatype

```
  [LAMBDA (obj)                                               (* bvm%: " 4-NOV-83 18:43")
    (PROG ((DTMAC (FASSOC (TYPENAME obj)
                          DT.EDITMACROS))
           newObj source installSourceFn changedFlg)
          (DECLARE (SPECVARS changedFlg))
          (OR DTMAC (RETURN (INSPECT obj)))

          (* CADR is a function which gets a list structure source for the datatype.
          CADDR is a function which installs the source back in the dataType.
          It is called when the source has been changed in the editing.)

          (COND
             ((NULL (SETQ source (APPLY* (CADR DTMAC)
                                         obj)))               (* If this fn returns NIL, we assume it has done any desired
                                                              editing itself)
              (RETURN)))
          (SETQ installSourceFn (CADDR DTMAC))
      LP  [SETQ source (EDITE source NIL obj (TYPENAME obj)
                              (FUNCTION (LAMBDA NIL
                                          (SETQ changedFlg T]
          [COND
             ((NOT changedFlg)
              (RETURN))
             ((NLSETQ (SETQ newObj (OR (APPLY* installSourceFn obj source)
                                       obj)))
              (RETURN (DEDITZAPCAR (TOPSELECTION)
                             newObj]
          (PROMPTPRINT "Error in datatype edit source")
          (GO LP))
    (DEDITReprint])

)

(ADDTOVAR DT.EDITMACROS )

;; Structure changing
```

```
(DEFINEQ

(SETPTRTO
  [LAMBDA (X Y)                                                    (* bas%: "12-Sep-84 16:25")
    (PROG (XM BK TEM)
          (COND
             ((NOT (SETQ XM (GETME4 X)))
              (CANT "Already deleted!"))
             ([SETQ TEM (GETLEFT XM (SETQ BK (GETMEBP XM]
              (DEDITZAPCDR TEM Y))
             ((fetch BP of BK)
              (DEDITZAPCAR BK Y))
             ((NLISTP Y)
              (CANT "Delete last list element"))
             (T [DEDITZAPBOTH X (CAR Y)
                       (COND
                          ((EQ X (CDR Y))
                           (RPLNODE2 Y X))
                          (T (CDR Y]
                (SETQ Y X)))
          (RETURN Y])

(DEDITCONS
  [LAMBDA (A D BROTHER)                                            (* bas%: "25-MAR-83 17:12")
    (fetch TAIL of (DUMMYMAPENTRY (CONS A D)
                          (GETMEBP (OR (GETME4 BROTHER)
                                       (CANT "Invalid target"])

(DEDITZAPCAR
  [LAMBDA (M A)                                                    (* bas%: " 2-MAR-83 15:38")
    (DEDITZAPBOTH M A (CDR (OR (LISTP M)
                               (fetch TAIL of M])

(DEDITZAPCDR
  [LAMBDA (M D)                                                    (* bas%: "25-JUL-82 16:23")
    (DEDITZAPBOTH M (CAR (OR (LISTP M)
                             (fetch TAIL of M)))
          D])

(DEDITZAPNODE
  [LAMBDA (M C)                                                    (* bas%: "27-JUL-81 04:48")
    (DEDITZAPBOTH M (CAR C)
          (CDR C])

(DEDITZAPBOTH
  [LAMBDA (CC A D ENT)                                             (* bas%: "18-Mar-84 15:19")
                                                                   (* ALL edit changes go through this function.)
    (COND
       [[SETQ ENT (COND
                     [(type? DEDITMAP CC)
                      (PROG1 CC
                          (SETQ CC (fetch TAIL of CC)))]
                     (T (GETME4 CC]
        (COND
           ((fetch BP of ENT))
           ((BUFSELP ENT))
           ((AND (EQ D (CDR CC))
                 (LISTP (CAR CC))
                 (LISTP A))
            (SETQ CC (CAR CC))

           (* We cant effect the dummy CONS held onto by the editor as that wont be seen by someone holding the defn
           (old EDIT just took error here) Here we try to oblige by sliding down into the first cell of the defn But we have to remove any
           pointers that the new CAR or CDR might have to the original cell, lest we create a cycle.)

           (SETQ D (COPYOUTCONS (CDR A)
                                CC))
           (SETQ A (COPYOUTCONS (CAR A)
                                CC)))
           (T (CANT "Alter top")))
        [COND
           ((DPCDRSEL ENT)
            [SETQ CC (LAST (fetch SELEXP of (fetch BP of ENT]          (* Real CONS)
            (SETQ D (COND
                       ((NEQ A (CDR CC))
                        A)
                       (T D)))
            (SETQ A (CAR CC))
            (PROG ((V (DOTTEDEND D)))
                  (COND
                     (V (DEDITFZAP (fetch TAIL of ENT)
```

```
                              V V))
                        (T (PUTHASH (fetch TAIL of (fetch BP of ENT))
                                   NIL \DEDITDPHASH)
                           (PUTHASH (fetch TAIL of ENT)
                                   NIL \DEDITMEHASH]
       (AND EDITSMASHUSERFN (APPLY* EDITSMASHUSERFN CC (GETEDITCHAIN ENT)))
       (COND
          ((DEDITFZAP CC A D)
           [PROG [(TEM (CDR (WINDOWPROP (fetch PDSP of ENT)
                                        'DEDITCHANGES]          (* Undoably smashes EDITCHANGES from call in which change
            is being made, unless already set)
                 (OR (NOT TEM)
                     (CAR TEM)
                     (DEDITFZAP TEM T (CDR TEM]
           (AND CHANGESARRAY (DEDITZAPCHANGES ENT))           (* A smashed cell is always changed)
           (for (E _ ENT) by (fetch BP of E) while E do (DEDITZAPCLISP (fetch SELEXP of E]
      (T (AND EDITSMASHUSERFN (APPLY* EDITSMASHUSERFN CC (LIST CC)))
          (DEDITFZAP CC A D])
```

(DEDITFZAP

```
[LAMBDA (CC A D)                                              (* bas%: "18-Mar-84 15:11")
```

(* Smashes cons CC and makes UNDOLST entry but uses no other context.
Used for making changes to editor structures sauch as the undo list itself)

```
   (PROG ((OA (CAR CC))
          (OD (CDR CC)))                                      (* Dont smash EQ values. Slow b/c of refcnts and clutters up
                                                              UNDOLST)
         (RETURN (AND (COND
                         ((EQ D OD)
                          (AND (NEQ A OA)
                               (FRPLACA CC A)))
                         ((EQ A OA)
                          (FRPLACD CC D))
                         (T (RPLNODE CC A D)))
                      (push UNDOLST1 (CONS CC (CONS OA OD])
```

(DEDITZAPCLISP

```
[LAMBDA (CC)                                                  (* bas%: "30-MAR-83 23:01")
```

(* Deletes CLISP translation. Not made part of the edit event, because of the possibility of the user performing two changes, and then undoing the first, which would then restore the translation, even though it no longer corresponds to the untranslated and changed CLISP.)

```
   (COND
      ((NLISTP CC))
      [(AND CLISPTRANFLG (EQ CLISPTRANFLG (CAR CC)))
       (COND
          ((LISTP (CDDR CC))
           (/RPLNODE2 CC (CDDR CC)))
          (T                                                  (* CLISP% used to translate an atom e.g.
                                                              QLISP does this.)
             (SHOULDNT]
      ((AND CLISPARRAY (GETHASH CC CLISPARRAY))
       (/PUTHASH CC NIL CLISPARRAY])
```

(DEDITZAPCHANGES

```
[LAMBDA (ME)                                                  (* bas%: "18-OCT-81 22:29")
   (COND
      ((for (I _ ME) by (fetch BP of I) while I never (GETHASH (fetch TAIL of I)
                                                               CHANGESARRAY))
       [push UNDOLST1 (CONS 'LISPXHIST (LIST (LIST '/PUTHASH (fetch TAIL of ME)
                                                   (GETHASH (fetch TAIL of ME)
                                                            CHANGESARRAY)
                                                   CHANGESARRAY]
```

(* Done this way for efficiency rather than going through editcom1 since we know what to undosave.)

```
       (PUTHASH (fetch TAIL of ME)
              ATM CHANGESARRAY])
```

(DEDITMOVETAILDOWN

```
[LAMBDA (C NUTAIL)                                            (* bas%: "12-Sep-84 14:41")
```

(* This moves C's current CDR to the end of the list which is its current CAR and replaces that CDR which it has just moved with NUTAIL. Order of moves helps simplify REPP)

```
   (DEDITZAPCDR (LAST (THELIST (CAR C)))
          (PROG1 (CDR C)
                 (DEDITZAPCDR C NUTAIL])
```

(DUNDOEDITL

```
  [LAMBDA (SS ULST ULST0)                                          (* lmm "26-Jul-86 23:35")
    (PROG (UNDOLST1 WAI)
          (for X on ULST until (EQ X ULST0) do (DUNDOEDITCOM (CAR X)) when (CAR X))
          (OR UNDOLST1 (SHOULDNT))                                 (* Must have some changes to undo)
          [bind TMP for I in ULST when [for J in (CDDDR I) thereis (SETQ TMP (WHICHEDITW (CAR J]
             do (AND (SETQ TMP (WINDOWPROP TMP 'DEDITWHOAMI))
                     (FMEMB (CADR TMP)
                            FILEPKGTYPES)
                     (MARKASCHANGED (CAR TMP)
                                    (CADR TMP]
          (DEDITFZAP ULST (CAR ULST0)
                 (CDR ULST0))                                      (* So undo can be UNDOne.)
          (COND
             (LISPXHIST (UNDOSAVE [LIST 'DUNDOEDITL SS (LIST (CONS T (CONS SS UNDOLST1]
                                  LISPXHIST])
```

(DUNDOEDITCOM

```
  [LAMBDA (X FLG)                                                  (* bas%: "12-Feb-84 21:25")
                                                                   (* If FLG is T, name of command is printed.)
    (COND
       ((NLISTP X)
        (CANT "Garbage on DEDIT UNDO list")                        (* Used to elseif (AND (CADR X)
                                                                   (NOT (SAMEEXPR \DSPRINTBP
                                                                   (fetch TOPELT of (CADR X))))) then
                                                                   (* The saved \DEDITSELECTIONS was not from the edit
                                                                   expression) (CANT "UNDO on different expression"))
        )
       ((CAR X)
        (DUNDOEDITCOM1 X)                                          (* else has been undone before, dont UNDO it again.)
        ))
    (COND
       (FLG (SETQ \DEDITSELECTIONS (CADR X))
            (printout PROMPTWINDOW T (OR (CAR X)
                                         "Already")
                      " undone.")))
    (DEDITFZAP X NIL (COPYCONS X))

          (* Marks X so UNDO will skip it in future. UNDOing this UNDO will unmark it)

    T])
```

(DUNDOEDITCOM1

```
  [LAMBDA (C)                                                      (* bas%: "21-MAR-83 19:43")

          (* Takes a single entry on UNDOLST, i.e. list of the form (command-name \DEDITSELECTIONS . UNDOLST1) and maps
          down the UNDOLST1 portion performing the corresonding DEDITSMASHes.)

    (for X in (CDDR C) do (SELECTQ (CAR X)
                                   (GROUPED

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

                                            (for X in (CDR X) do (DUNDOEDITCOM1 X)))
                                   (LISPXHIST (EDITBLOCKCALL EDITCOM1 (CDR X)))
                                   (DEDITZAPNODE (CAR X)
                                          (CDR X])

)

;; Selection code.  Select expressions or from the command menu

(DEFINEQ
```

(DEDITSLCTLP

```
  [LAMBDA (CDS)                                                    (* mjs "26-Mar-86 16:27")
                                                                   (* Does selections until a command is given)
    (RESETLST
        (RESETSAVE (DEDITUSER T))
        (RESETSAVE \DEDITALLOWSELS T)
        [CAR (ERSETQ (bind CMD do (WAIT.FOR.TTY)
                                  (SETEDITMENU (COND
                                                  ((KEYDOWNP 'TAB)
                                                   NIL)
                                                  (T CDS)))
                                  [COND
                                     ((NOT (\SYSBUFP))
                                      (SETQ CMD (READEDITMENU)))
                                     ((EQ (\PEEKSYSBUF)
                                          (CHARCODE TAB))
                                      (\GETSYSBUF)                 (* Flush TAB char)
                                      )
                                     [(SETQ CMD (DODEDITTYPEDCOM (GETEBUF CDS]
                                     (T (SELECTKEYS (GETEBUF CDS]
                                  (AND CMD (RETURN CMD))
                                  (BLOCK])])
```

(DEDITUSER

```
[LAMBDA (DS)                                                    (* bas%: "12-Apr-84 20:17")
  (FLIPSELS)
  (SETCURSOR (COND
                (DS DEFAULTCURSOR)
                (T WAITINGCURSOR)))
  (NOT DS])
```

(SELECTKEYS

```
[LAMBDA (W)                                                     (* mjs "26-Mar-86 16:19")
  (PROG ((LINE (DEDITREADLINE NIL W)))
        (SHADEIFNOTBUF (NXTSELECTION T)
               SECSHADE)                                        (* Push shading)
        (SHADEIFNOTBUF (TOPSELECTION T)
               SWITCHSHADE)
        (SHADESELECTION (SETUPDEDITW W (PUSHSELECTION (LIST LINE)))
               PRIMSHADE])
```

(DODEDITTYPEDCOM

```
[LAMBDA (W)                                                     (* mjs "26-Mar-86 16:19")
  (bind (C _ (\PEEKSYSBUF)) for I in DEDITTYPEINCOMS do (COND
                                                          ((EQ C (CONTROLCODE (CAR I)))
                                                           (\GETSYSBUF)
                                                           (printout W (CADR I)
                                                                  ": ")
                                                           (RETURN (CONS (CADR I)
                                                                         (CONS (CADDR I)
                                                                               (DEDITREADLINE T W])
```

(DEDITREADLINE

```
[LAMBDA (ASLIST W)                                              (* mjs "26-Mar-86 16:19")
```

(* Read a line of input from T. This is like the grunge that goes on inside of LISPX, figuring out where the line ends...)

```
  (RESETLST
      (RESETSAVE (TTYDISPLAYSTREAM W))
      (RESETSAVE \DEDITALLOWSELS NIL)
      [PROG ((FIRSTITEM (APPLY* LISPXREADFN T T))
             CH LINE)
            (RETURN (COND
                        ((AND (LISTP FIRSTITEM)
                              (OR (SYNTAXP (SETQ CH (CHCON1 (LASTC T)))
                                           'RIGHTPAREN T)
                                  (SYNTAXP CH 'RIGHTBRACKET T)))
```

(* A list is the first thing typed. Usually, there is no more, but you could get a list from an "atomic" form if it had a read macro that turned it into a list)

```
                         (COND
                            (ASLIST (LIST FIRSTITEM))
                            (T FIRSTITEM)))
                        ((OR (CDR (SETQ LINE (READLINE T (LIST FIRSTITEM)
                                                       T)))
                             ASLIST)                            (* There was more, so return whole list)
                         LINE)
                        (T                                      (* Single atom)
                           FIRSTITEM])])
```

(SHADEIFNOTBUF

```
[LAMBDA (X TXT)                                                 (* bas%: "13-MAR-83 19:59")
  (AND X (SETQ X (GETSELMAP X))
       (NOT (BUFSELP X))
       (SHADESELECTION X TXT])
```

(DEDITBUTTONFN

```
[LAMBDA (W)                                                     (* bas%: " 1-Apr-84 15:34")
  (TOTOPW W)                                                    (* Bring it up, if nothing else)
  (COND
     ((SHIFTSELECTKEYS)
      (SELECTREAD W))
     (\DEDITALLOWSELS (SELECTELEMENT W])
```

(DEDITRIGHTBUTTONFN

```
[LAMBDA (W)                                                     (* bas%: " 1-Apr-84 15:31")
  (TOTOPW W)                                                    (* Bring it up, if nothing else)
  (COND
     ((AND \DEDITALLOWSELS (INWINDOW W))
      (SELECTTREE W))
     (T (DOWINDOWCOM W])
```

(DEDITWINDOWENTRYFN

```
[LAMBDA (W)                                                    (* bas%: " 1-Apr-84 15:19")
                                                               (* Shift the tty process if not a shift select and not currently tty
                                                               proc)
   (TOTOPW W)                                                  (* Bring it up, if nothing else)
   (COND
      ((SHIFTSELECTKEYS)
       (SELECTREAD W))
      (T (GIVE.TTY.PROCESS W])
```

(SELECTELEMENT

```
[LAMBDA (DS)                                                   (* bas%: "24-MAR-83 16:01")
   (bind N M (TE _ (GETSELMAP (TOPSELECTION T)))
         (NE _ (GETSELMAP (NXTSELECTION T))) until (SELECTDONE DS)
      do (AND (SETQ M (SEARCHMAP DS))
              (LASTMOUSESTATE MIDDLE)
              (SETQ M (fetch BP of M)))
         (COND
            ((EQ M N))
            (T (COND
                  ((AND N M))
                  (T (SHADESELECTION NE SECSHADE)              (* Virtual push/pop)
                     (SHADESELECTION TE SWITCHSHADE)))
               (SHADESELECTION N PRIMSHADE)
               (SHADESELECTION M PRIMSHADE)
               (SETQ N M)))
      finally (AND M (PUSHSELECTION (fetch TAIL of M])
```

(SELECTREAD

```
[LAMBDA (DS)                                                   (* bvm%: " 4-Jun-86 18:48")
   (bind M N while (SHIFTSELECTKEYS) do (until (SELECTDONE DS) do (AND (SETQ M (SEARCHMAP DS))
                                                                      (LASTMOUSESTATE MIDDLE)
                                                                      (SETQ M (fetch BP of M)))
                                                                (COND
                                                                   [(AND N M)
                                                                    (COND
                                                                       ((EQ M N))
                                                                       (T (SHADESELECTION N READSHADE)
                                                                          (SHADESELECTION M READSHADE]
                                                                   (T (SHADESELECTION (OR N M)
                                                                             READSHADE)))
                                                                (SETQ N M))
      finally (COND
                 (M (SHADESELECTION M READSHADE)
                    (WITH-READER-ENVIRONMENT (WINDOWPROP DS 'READER-ENVIRONMENT)
                        (BKSYSBUF (fetch SELEXP of M)
                               T)
                        [COND
                           ((LISTP (fetch SELEXP of M)))
                           (T (BKSYSCHARCODE (CHARCODE SPACE])])
```

(SELECTTREE

```
[LAMBDA (DS)                                                   (* bas%: " 1-Apr-84 15:17")
   (bind (OT _ (GETME4 (TOPSELECTION)
                       T))
      until (SELECTDONE DS) do (SWITCHANDSHADE (FINDLCA OT (SEARCHMAP DS])
```

(SEARCHMAP

```
[LAMBDA (PDS)                                                  (* bas%: "20-Apr-84 14:37")
   (PROG (L S (E (GETDEDITMAP PDS))
            (LX (LASTMOUSEX PDS))
            (LY (LASTMOUSEY PDS)))
         [while E until (AND (WITHINME E LX LY)
                             (OR [NOT (SETQ L (LISTP (fetch SELEXP of (SETQ S E]
                                 (ONAPARENP E LX LY)))
            do

         (* The until clause is true if either E covers mouse and has no descendents or we're on a paren)
                                                               (* Either pending tail or embedded descendents to search)
               [COND
                  [(NOT (SETQ E (GETME4 L S]
                  ((HASASBP E S))
                  (T (REPP S)                                  (* Substructure has been smashed.
                                                               Reprint and start over.)
                     (SETQ E (GETME4 (fetch TAIL of S)
                                     T))
                     (SETQ S (fetch BP of E))
                     (SETQ L (fetch TAIL of E]
               (SETQ L (CDR (LISTP L]
         (RETURN E])
```

```
(WITHINME
  [LAMBDA (E X Y)                                          (* bas%: "30-MAR-83 14:24")
    (PROG [(FA (FONTPROP (fetch FNT of E)
                         'ASCENT))
           (FD (FONTPROP (fetch FNT of E)
                         'DESCENT]
          (RETURN (COND
                     ((IGREATERP Y (IPLUS FA (fetch STARTY of E)))
                      NIL)
                     [(IGEQ Y (IDIFFERENCE (fetch STARTY of E)
                                           FD))
                      (AND (IGEQ X (fetch STARTX of E))
                           (OR (ILESSP X (fetch STOPX of E))
                               (NEQ (fetch STARTY of E)
                                    (fetch STOPY of E]
                     ((ILESSP Y (IDIFFERENCE (fetch STOPY of E)
                                             FD))
                      NIL)
                     [(IGREATERP Y (IPLUS FA (fetch STOPY of E]
                     (T (ILESSP X (fetch STOPX of E])

(ONAPARENP
  [LAMBDA (E X Y)                                          (* bas%: "30-MAR-83 14:24")
    (PROG ((EF (fetch FNT of E)))
          (RETURN (OR [AND (ILESSP X (fetch LPEND of E))
                           (IGEQ Y (IDIFFERENCE (fetch STARTY of E)
                                                (FONTPROP EF 'DESCENT]
                      (AND (IGEQ X (fetch RPSTART of E))
                           (ILESSP Y (IPLUS (fetch STOPY of E)
                                            (FONTPROP EF 'ASCENT])

(SELECTDONE
  [LAMBDA (PDS)                                            (* bas%: "28-JUL-82 22:42")
    (OR (MOUSESTATE UP)
        (NOT (INWINDOW PDS])

(INWINDOW
  [LAMBDA (DS)                                             (* bas%: "27-AUG-82 12:38")
    (INSIDE? (DSPCLIPPINGREGION NIL DS)
             (LASTMOUSEX DS)
             (LASTMOUSEY DS])

(FINDLCA
  [LAMBDA (S1 S2)                                          (* bas%: " 1-Apr-84 15:17")
    (COND
       ((NOT S2)
        S1)
       ((EQ (fetch PDSP of S1)
            (fetch PDSP of S2))
        (for old S1 while S1 by (fetch BP of S1) thereis (DOMINATE? S1 S2])

(DOMINATE?
  [LAMBDA (SUP SUB)                                        (* bas%: " 4-Apr-84 13:06")
    (OR (EQ SUP SUB)
        (PROG [(S1 (OR (MAPENTRYP SUP)
                       (GETME4 SUP)))
               (S2 (OR (MAPENTRYP SUB)
                       (GETME4 SUB]
              (RETURN (COND
                         ((AND S1 S2)
                          (for old S2 by (fetch BP of S2) while S2 thereis (EQ S1 S2)))
                         (T (for I on (CAR (LISTP SUP)) thereis (DOMINATE? I SUB])

)

(ADDTOVAR DEDITTYPEINCOMS
          [F Find (NLAMBDA (TGT)
                       (PUSHSELECTION (LIST TGT))
                       (DEDITSwap)
                       (DEDITFind]
          [S Substitute (NLAMBDA (OLD NEW)
                              (DEDITEditCom (LIST 'R OLD NEW]
          [Z EditCom (NLAMBDA EC (DEDITEditCom EC])

(PUTPROPS DEDITTYPEINCOMS VARTYPE ALIST)

;; Handling the selection stack

(DEFINEQ

(POPSELECTION
```

```
  [LAMBDA NIL                                                    (* bas%: "21-MAR-83 19:43")
    (PROG1 (TOPSELECTION)
           (pop \DEDITSELECTIONS])
```

(PUSHSELECTION

```
  [LAMBDA (S)                                                    (* bas%: "21-MAR-83 19:43")
    (push \DEDITSELECTIONS S)
    S])
```

(NXTSELECTION

```
  [LAMBDA (NOERR)                                                (* bas%: "24-MAR-83 15:52")
    (OR (fetch NXTELT of \DEDITSELECTIONS)
        (AND (NOT NOERR)
             (CANT "No second selection"])
```

(TOPSELECTION

```
  [LAMBDA (NOERR)                                                (* bas%: "24-MAR-83 15:52")
    (OR (fetch TOPELT of \DEDITSELECTIONS)
        (AND (NOT NOERR)
             (CANT "Too few selections"])
```

(SWITCHANDSHADE

```
  [LAMBDA (NU)                                                   (* bas%: " 1-Apr-84 15:29")
                                                                 (* Like a POP/PUSH sequence but no CONS)
    (COND
       [(OR (NOT NU)
            (EQ (fetch TAIL of NU)
                (TOPSELECTION T]
       (T (SHADESELECTION (GETME4 (TOPSELECTION T)
                                  T)
                          PRIMSHADE)
          (replace TOPELT of \DEDITSELECTIONS with (fetch TAIL of NU))
          (SHADESELECTION NU PRIMSHADE])
```

(SHADESELECTION

```
  [LAMBDA (S SHADE)                                              (* rrb "13-Feb-86 16:45")
    (AND S (SHADESELECTION1 S SHADE])
```

(SHADESELECTION1

```
  [LAMBDA (S TXT)                                                (* bvm%: "22-May-86 12:49")
    (LET ((START (fetch STARTY of S))
          (STOP (fetch STOPY of S)))
         (COND
            ((EQ START STOP)                                     (* "All on one line.  Last clause handles this in general, but test
                                                                 common case here for efficiency")
             (SHADESELECTION2 S START (fetch STARTX of S)
                    (fetch STOPX of S)
                    TXT))
            ((LISTP (fetch SELEXP of S))                         (* "Shade the parens and every element")
             (SHADESELECTION2 S START (fetch STARTX of S)
                    (fetch LPEND of S)
                    TXT)
             [for E on (fetch SELEXP of S) do (SHADESELECTION1 (GETME4 E S)
                                                               TXT)
                finally (COND
                           (E                                    (* Dotted pair)
                              (SHADESELECTION1 (GETME4 E S)
                                               TXT]
             (SHADESELECTION2 S STOP (fetch RPSTART of S)
                    (fetch STOPX of S)
                    TXT))
            (T                                                   (* "A non-list spanning more than one line, probably a string.  We
                                                                 don't know where the internal margins are, so be conservative")
               (LET* [(DS (fetch PDSP of S))
                      [LEFT (COND
                                [(fetch LONGSTRINGP of S)
                                 (fetch STARTX of (COND
                                                     ((fetch LONGSTRING1MARGINP of S)
                                                      S)
                                                     (T (fetch BP of S]
                                (T (DSPLEFTMARGIN NIL DS]
                       (RIGHT (COND
                                 ((fetch LONGSTRINGSYMMETRICP of S)
                                  (IDIFFERENCE (DSPRIGHTMARGIN NIL DS)
                                               LEFT))
                                 (T (DSPRIGHTMARGIN NIL DS]
                     (for I from START by (IMINUS (FONTPROP (fetch FNT of S)
                                                            'HEIGHT))
                        to STOP do (SHADESELECTION2 S I (COND
                                                           ((EQ I START)
                                                            (fetch STARTX of S))
```

```
                                                    (T LEFT))
                                            (COND
                                               ((EQ I STOP)
                                                (fetch STOPX of S))
                                               (T RIGHT))
                                            TXT])
```

(SHADESELECTION2

```
  [LAMBDA (S CY SX EX SHADE)                                    (* bas%: "13-JUL-82 10:02")
    (BITBLT NIL NIL NIL (fetch PDSP of S)
           SX
           (IDIFFERENCE CY (ADD1 LINETHICKNESS))
           (IDIFFERENCE EX SX)
           LINETHICKNESS
           'TEXTURE
           'INVERT SHADE])
```

(OVERLAPSELBAND

```
  [LAMBDA (S H L)                                               (* bas%: "26-Mar-84 15:17")
    (OVERLAP (SUB1 (fetch STARTY of S))
           (IDIFFERENCE (fetch STOPY of S)
                  (ADD1 LINETHICKNESS))
           H L])
```

(PUSHEDITCHAIN

```
  [LAMBDA (C)                                                   (* bas%: "30-MAR-83 22:19")
    [PUSHSELECTION (PROG ((X (MAKESELCHAIN C)))
                         (RETURN (COND
                                    ((MAPENTRYP X)
                                     (fetch TAIL of X))
                                    (T C]
    (DEDITCenter T])
```

(MAKESELCHAIN

```
  [LAMBDA (LST)                                                 (* bas%: " 5-Apr-84 21:03")
```

(* Makes dummy map entries until the whole chain is linked into an extant map.
This is necessary so subsequent commands from a Multiple can find their way around)

```
    (PROG (TMP)
          (DECLARE (USEDFREE LASTAIL))
          (COND
             [(CDR (THELIST LST))
              (SETQ TMP (OR [COND
                                ((LISTP (CAR LST))
                                 (TAILP (CAR LST)
                                        (CADR LST)))
                                (T (OR (TAILP LASTAIL (CADR LST))
                                       (EQ (CAR LST)
                                           (DOTTEDEND (CADR LST]
                            (FMEMB (CAR LST)
                                   (CADR LST))
                            (CANT "Inconsistent EDIT chain")))
              (RETURN (OR (GETME4 TMP)
                          (DUMMYMAPENTRY TMP (MAKESELCHAIN (CDR LST]
             (T (SETQ TMP (GETME4 (CAR LST)))
                (RETURN (AND (MAPENTRYP TMP)
                             (GETMEBP TMP])
```

(PUSHINTOBUF

```
  [LAMBDA (V)                                                   (* bas%: " 4-MAR-83 12:23")
    (AND V (PUSHSELECTION V])
```

(DUMMYMAPENTRY

```
  [LAMBDA (E B)                                                 (* bas%: "12-Sep-84 10:46")
                                                                (* Dummys are marked by having EQ startx and stopx)
    (MAKEMAPENTRY (OR (LISTP E)
                      (MAKEDOTPTAIL E B))
           B 0 0 0 0 (fetch F# of B])
```

(FLIPSELS

```
  [LAMBDA NIL                                                   (* bas%: "26-Mar-84 18:21")
                                                                (* Turns selections on or off across possible movement)
    (PROG [(TM (FIXUPSEL (TOPSELECTION T]
          (SHADESELECTION (FIXUPSEL (NXTSELECTION T)
                                    (BUFSELP TM))
                 SECSHADE)
          (SHADESELECTION TM PRIMSHADE])
```

(FLIPSELSIN

```
  [LAMBDA (DS H L)                                          (* bas%: " 4-Apr-84 13:18")
                                                            (* Turns selections on or off across possible movement)
    (SETQ DS (WINDOWPROP DS 'DSP))
    (PROG (S)
          (AND (SETQ S (GETME4 (NXTSELECTION T)))
               (EQ DS (fetch PDSP of S))
               (OVERLAPSELBAND S H L)
               (SHADESELECTION (UNPURGEDP S)
                      SECSHADE))
          (AND (SETQ S (GETME4 (TOPSELECTION T)))
               (EQ DS (fetch PDSP of S))
               (OVERLAPSELBAND S H L)
               (SHADESELECTION (UNPURGEDP S)
                      PRIMSHADE])
```

(FIXUPSEL

```
  [LAMBDA (X BUFBUSY)                                       (* bas%: "24-Jun-84 17:48")
                                                            (* Returns a new selection if X is not OK)
    (AND X (OR (GETSELMAP X)
               (AND (PROG1 (UNZORCHME (GETME4 X))
```

(* GETME4 and thus the UNZORCHME only succeeds after GETSELMAP has failed if X's map has been invalidated.
Usually the result is that X should be flushed into the edit buffer.
However, if X is invalid b/c the whole window has been ZORCHed
(by a background MARKASCHANGED e.g.) then we reestablish the whole window and try again)

```
                          )
                    (GETSELMAP X))
               (AND (NOT BUFBUSY)
                    (SETUPDEDITW (GETEBUF (TOPEDITW))
                           (NEWSELFOR X])
```

(NEWSELFOR

```
  [LAMBDA (X)                                               (* bas%: "24-MAR-83 16:03")
    (PROG ((Y (CONS (COPY (CAR X))
                    NIL)))
          (COND
             ((EQ X (TOPSELECTION T))
              (replace TOPELT of \DEDITSELECTIONS with Y))
             ((EQ X (NXTSELECTION T))
              (replace NXTELT of \DEDITSELECTIONS with Y))
             (T (SHOULDNT)))
          (RETURN Y])

)

;; Initializing and flushing edit windows

(DEFINEQ

(ACTIVEEDITW
  [LAMBDA (W ONFLG)                                         (* lmm " 9-Jul-85 16:30")
    (WINDOWPROP W 'BUTTONEVENTFN (AND ONFLG (FUNCTION DEDITBUTTONFN)))
    [WINDOWPROP W 'RIGHTBUTTONFN (COND
                                    (ONFLG (FUNCTION DEDITRIGHTBUTTONFN))
                                    (T (FUNCTION DOWINDOWCOM]
    (WINDOWPROP W 'RESHAPEFN (AND ONFLG (FUNCTION DEDITRESHAPEFN)))
    (WINDOWPROP W 'REPAINTFN (AND ONFLG (FUNCTION DEDITREPAINTFN)))
    (WINDOWPROP W 'SCROLLFN (AND ONFLG (FUNCTION SCROLLBYREPAINTFN)))
    (WINDOWPROP W 'PROCESS (THIS.PROCESS))                  (* So that bugging in this window can switch tty to us)
    [WINDOWPROP W 'WINDOWENTRYFN (COND
                                    (ONFLG (FUNCTION DEDITWINDOWENTRYFN))
                                    (T (FUNCTION GIVE.TTY.PROCESS]
    (DSPSCROLL (COND
                  (ONFLG 'OFF)
                  (T T))
           W)                                               (* Buffer can get this turned on)
    W])

(FINDEDITW
  [LAMBDA (NAME TYPE)                                       (* bas%: "12-Sep-84 22:24")
    (for I in \DEDITWINDOWS thereis (SAMEEDITW I NAME TYPE])

(GETEDITW
  [LAMBDA (ATM TYPE)                                        (* rrb " 2-Oct-85 18:44")
    (SELECTQ TYPE
         (NIL (OR ATM (SETQ ATM (CONCAT " ")))              (* A unique, but invisible tag)
              (SETQQ TYPE expression))
         (PROP (SETQQ TYPE FNS))
         NIL)
    (PROG [(W (OR (FINDEDITW ATM TYPE)
                  (MAKEEDITW ATM TYPE]
```

```
          (RESETSAVE NIL (LIST 'UNDEDITW (push \DEDITWINDOWS W)))

          (* make this process be the process for this window so that clicking in it will give the tty to this Dedit.)

          (WINDOWPROP W 'PROCESS (THIS.PROCESS))
          (RETURN (WINDOWPROP W 'DSP])
```

(GETDEDITDEF4

```
  [LAMBDA (W)                                                    (* bas%: "10-Mar-84 11:55")
    (PROG [NAME (TYPE (WINDOWPROP W 'DEDITWHOAMI]
          (RETURN (AND (SETQ NAME (CAR TYPE))
                       (LITATOM NAME)
                       (SETQ TYPE (CADR TYPE))
                       (NEQ TYPE 'expression)
                       (GETDEF NAME TYPE NIL '(NOCOPY NOERROR])
```

(MAKEEDITW

```
  [LAMBDA (NAME TYP)                                             (* rrb " 2-Oct-85 18:44")
    (PROG [(W (COND
                  ((TOPEDITW)
                   (WINDOWPROP (TOPEDITW)
                          'DEDITCACHED NIL))
                  (T (WINDOWP DEditWindow]
          (DECLARE (USEDFREE EDITCHANGES))
          (AND (COND
                  [(NOT W)
                   (SETQ W (CREATEW NIL (NAMEOFEDITW NAME TYP]
                  ((NOT (SAMEEDITW W NAME TYP))
                   (CLEARW W)
                   (WINDOWPROP W 'TITLE (NAMEOFEDITW NAME TYP))
                   T))
               (WINDOWPROP W 'DEDITWHOAMI (LIST NAME TYP)))
          (WINDOWPROP W 'DEDITCHANGES EDITCHANGES)               (* Associates changes with changed structure)
          (RETURN W])
```

(NAMEOFEDITW

```
  [LAMBDA (NAME TYPE)                                            (* bas%: "30-MAR-83 18:41")
    (CONCAT "DEdit of " (SELECTQ TYPE
                               (FNS "function")
                               (PROPS (COND
                                         [(CADR (LISTP NAME))
                                          (PROG1 (CONCAT (CADR NAME)
                                                         " property of ")
                                              (SETQ NAME (CAR NAME)))]
                                         (T "property list of")))
                               (VARS (COND
                                        [(AND (STREQUAL (SUBSTRING NAME -4 -1)
                                                        "COMS")
                                              (HASDEF (SUBSTRING NAME 1 -5)
                                                      'FILE))
                                         (PROG1 "filecoms for file"
                                             (SETQ NAME (SUBSTRING NAME 1 -5)))]
                                        (T "variable")))
                               TYPE)
            " " NAME])
```

(PURGEW

```
  [LAMBDA (W DONTCLR)                                            (* rmk%: "13-Sep-84 16:49")
    [PROG [(WDS (COND
                   ((WINDOWP W)
                    (WINDOWPROP W 'DSP))
                   (T (PROG1 W
                             (SETQ W (WFROMDS W)))]
          [COND
             ((EQ W DEditWindow)
              (CLRHASH \DEDITMEHASH)
              (CLRHASH \DEDITDPHASH))
             (T (MAPHASH \DEDITMEHASH (FUNCTION (LAMBDA (X Y)
                                                  (AND (EQ WDS (fetch PDSP of X))
                                                       (PUTHASH Y NIL \DEDITMEHASH]
          [for I to (ARRAYSIZE \DEDITDSPS) when (EQ WDS (ELT \DEDITDSPS I))
             do (RETURN (SETA \DEDITDSPS I (WINDOWPROP WDS 'REGION]
          (WINDOWPROP W 'EDITEXPR NIL)
          (COND
             (DONTCLR)
             (T (DSPTEXTURE WHITESHADE W)
                (DSPFONT DEFAULTFONT W)                          (* Font first to get CLEARW right)
                (CLEARW W)
                (MAKECPOSBE (DSPXPOSITION NIL W)
                       (CONSTANT (IDIFFERENCE MAX.SMALLP 1535))
                       W]
    W])
```

(MAKECPOSBE

```
[LAMBDA (X Y DS)                                        (* bas%: " 4-Apr-84 13:11")
  (PROG [(DX (IDIFFERENCE X (DSPXPOSITION NIL DS)))
         (DY (IDIFFERENCE Y (DSPYPOSITION NIL DS]
        (WXOFFSET (IMINUS DX)
               DS)
        (WYOFFSET (IMINUS DY)
               DS)
        (RELMOVETO DX DY DS])
```

(SAMEEDITW

```
[LAMBDA (W NAME TYPE)                                   (* bas%: "15-FEB-82 18:16")
  (PROG [(TMP (WINDOWPROP W 'DEDITWHOAMI]
        (RETURN (AND TMP (EQ NAME (CAR TMP))
                     (EQ TYPE (CADR TMP])
```

(SETUPDEDITW

```
[LAMBDA (W CONTENTS)                                    ; Edited 29-Aug-91 13:38 by jds
```

;; Set up the DEDIT editing window W to be editing CONTENTS.

;; Must bind several specials that control the pretty printer here, because this is where we can figure out the information:

;; FIRSTPOS = left margin

;; RMARGIN = right margin

;; FILEFLG = are we printing to a file? (No!)

```
  (LET ((FIRSTPOS (DSPLEFTMARGIN NIL *STANDARD-OUTPUT*))
        (RMARGIN (DSPRIGHTMARGIN NIL W))
        (FILEFLG NIL)
        (COMMENTCOL NIL))
       (DECLARE (SPECVARS FIRSTPOS RMARGIN FILEFLG COMMENTCOL))
       (PROG1 (SETDEDITMAP W CONTENTS)
              (ACTIVEEDITW W T])
```

(TOPEDITW

```
[LAMBDA NIL                                             (* bas%: "18-MAR-83 15:25")
  (CAR \DEDITWINDOWS])
```

(UNDEDITW

```
[LAMBDA (WDS)                                           (* bvm%: "22-May-86 12:46")

        (* * "Desensitizes DEDIT windows and removes surplus ones")

  (COND
     (\DEDITMNUW (WINDOWPROP \DEDITMNUW 'PROCESS NIL)
            (CLOSEW \DEDITMNUW)))
  (PROG [(W (WFROMDS (OR (CAR (LISTP WDS))
                         (SHOULDNT]
        (DECLARE (USEDFREE DEditLinger))
        (TAKEDOWN (WINDOWPROP W 'EDITBUF))
        (SETQ \DEDITBUFW NIL)
        [COND
           ((EQ WDS \DEDITWINDOWS)
            (SETQ \DEDITWINDOWS (CDR WDS)))
           (T (for I on \DEDITWINDOWS when (EQ WDS (CDR I)) do (RETURN (RPLACD I (CDDR I)))
              finally (SHOULDNT "DEDITDSPS tangled"]
        (COND
           [\DEDITWINDOWS (COND
                             ((FMEMB W \DEDITWINDOWS))
                             (T (WINDOWPROP W 'DEDITCACHED NIL)
                                                        (* "Discard my cache;  cache me on next")
                                (WINDOWPROP (TOPEDITW)
                                       'DEDITCACHED W)
                                (SETQ \DEDITBUFW (WINDOWPROP (TOPEDITW)
                                                             'EDITBUF))
                                (TAKEDOWN W]
           (T (COND
                 ((AND RESETSTATE (CADR (WINDOWPROP W 'DEDITCHANGES NIL)))
                  (ZORCHEDITW W)))
              (OR (WINDOWP DEditWindow)
                  (SETQ DEditWindow W))
              (WINDOWPROP W 'PROCESS NIL)
              (OR DEditLinger (CLOSEW W])
```

(WHICHEDITW

```
[LAMBDA (CC)                                            (* bas%: " 4-FEB-83 15:45")
  (bind SCR for TMP from (GETME4 CC) by (fetch BP of TMP) while TMP
     do (AND (SETQ SCR (EDITWINDOWP (fetch PDSP of TMP)))
             (RETURN SCR])
```

```
(ZORCHEDITW
  [LAMBDA (W)                                                    (* hdj "19-Jul-85 11:35")
    (AND W [PROG ((V (GETMAP? W)))
                 (COND
                    ((AND V (NOT (fetch BP of V)))
                     (replace BP of V with (create DEDITMAP
                                                   D# _ (fetch D# of V)))
                     (RETURN T]
         (ACTIVEWP (WFROMDS W))
         (PROGN (DSPTEXTURE CHANGEDSHADE W)
                (DSPFILL NIL CHANGEDSHADE 'PAINT W])

(ZORCHEDWP
  [LAMBDA (W)                                                    (* bas%: "11-Mar-84 22:33")
    (PROG [(WM (GETME4 (WINDOWPROP W 'EDITEXPR]                  (* ZORCHed windows have a dummy map in the BP of their
                                                                   EDITEXPR's map)
          (RETURN (AND WM (fetch BP of WM])

(UNZORCHME
  [LAMBDA (M)                                                    (* bas%: "11-Mar-84 23:15")
    (AND M (PROG ((W (fetch PDSP of M)))
                 (COND
                    ((ZORCHEDWP W)
                     (RETURN (SETDEDITMAP W (LIST (GETDEDITDEF4 W])

)

(RPAQ? DEditLinger T)

;; Manipulating the Edit menu

(DEFINEQ

(SETEDITMENU
  [LAMBDA (EW)                                                   (* bvm%: "30-May-86 16:04")
    (DECLARE (GLOBALVARS \DEDITCOMS))
    (PROG (MR X Y H W IMAGE)
          [SETQ MR (AND (WINDOWP \DEDITMNUW)
                        (WINDOWPROP \DEDITMNUW 'REGION]

          (* The WINDOWP check on \DEDITMNUW is b/c it can be a displaystream if user interrupts out of READEDITMENU in
          which case it must be rebuilt b/c of possible undone inversions)

          [COND
             (MR (SETQ W (fetch (REGION WIDTH) of MR))
                 (SETQ H (fetch (REGION HEIGHT) of MR)))
             (T (SETQ IMAGE (CACHEDEDITCOMS *DEDIT-MENU-COMMANDS*))
                (SETQ W (ITIMES 2 (SUB1 WBorder)))
                (SETQ H (IPLUS (BITMAPHEIGHT IMAGE)
                               (IMINUS (DSPLINEFEED NIL WindowTitleDisplayStream))
                               W))
                (SETQ W (IPLUS (BITMAPWIDTH IMAGE)
                               W]
          [COND
             [EW (PROG (ER)
                       (SETQ ER (WINDOWPROP EW 'REGION))
                       (SETQ X (fetch (REGION PRIGHT) of ER))
                       (SETQ Y (IDIFFERENCE (fetch (REGION PTOP) of ER)
                                            H]
             (T (GETMOUSESTATE)
                (SETQ X (IDIFFERENCE LASTMOUSEX WBorder))
                (SETQ Y (IDIFFERENCE LASTMOUSEY (WINDOWPROP \DEDITMNUW 'YOFFSET]
          (SETQ X (IMIN X (IDIFFERENCE SCREENWIDTH W)))
          [SETQ Y (IMAX 0 (IMIN Y (IDIFFERENCE SCREENHEIGHT H]
          [COND
             (MR (COND
                    [(AND (EQ X (fetch (REGION LEFT) of MR))
                          (EQ Y (fetch (REGION BOTTOM) of MR]
                    (T (MOVEW \DEDITMNUW X Y)))
                 (TOTOPW \DEDITMNUW))
             (T (PROG (NUR)
                      (SETQ NUR (create REGION
                                        LEFT _ X
                                        BOTTOM _ Y
                                        WIDTH _ W
                                        HEIGHT _ H))
                      [COND
                         ((DISPLAYSTREAMP \DEDITMNUW)
                          (SETQ \DEDITMNUW (WFROMDS \DEDITMNUW))
                          (WINDOWPROP \DEDITMNUW 'RESHAPEFN NIL)
                          (SHAPEW \DEDITMNUW NUR))
                         (T (SETQ \DEDITMNUW (CREATEW NUR 'EditOps]
                      (WINDOWPROP \DEDITMNUW 'RESHAPEFN 'DON'T))
                (BITBLT IMAGE 1 1 \DEDITMNUW 0 0 W H 'INPUT 'REPLACE)
                                                                 (* The 1,1 removes the menu border)
```

```
            (WINDOWPROP \DEDITMNUW 'IMAGE IMAGE)
            (WINDOWPROP \DEDITMNUW 'ITEMHEIGHT (FONTPROP MENUFONT 'HEIGHT))
            (WINDOWPROP \DEDITMNUW 'YOFFSET (IQUOTIENT H 2))
            (WINDOWPROP \DEDITMNUW 'REPAINTFN 'DEDITMENURESTORE]
      (WINDOWPROP \DEDITMNUW 'PROCESS (THIS.PROCESS))       (* Allow the menu window to also respond to tty switching)
      (RETURN \DEDITMNUW])
```

(CACHEDEDITCOMS

```
[LAMBDA (COMLIST)                                           (* bvm%: "30-May-86 16:20")
```

(* * "Builds a menu image from the commands in COMLIST. Sets arrays EDITMENU\COMS and EDITMENU\SUBS with elements in INVERSE order for convenience of READEDITMENU")

```
   (DECLARE (GLOBALVARS EDITMENU\COMS EDITMENU\SUBS))
   (LET* ((N (LENGTH COMLIST))
          (COMS (ARRAY N NIL NIL 0))
          (SUBMENUS (ARRAY N NIL NIL 0)))
         [for ITEM in COMLIST as J from (SUB1 N) by -1
            do (SETA COMS J (CONS (CAR ITEM)
                                  (CADR ITEM)))           (* The main item)
               (SETA SUBMENUS J (AND (CDDR ITEM)
                                     (create MENU
                                             ITEMS _ [for Q in (CDDR ITEM)
                                                        collect `(,(CAR Q)
                                                                  ',(CAR Q)
                                                                    ,@(CADR Q))
                                                                  ,@(CDDR Q]
                                             CENTERFLG _ T
                                             MENUOFFSET _ (create POSITION
                                                                  XCOORD _ -1
                                                                  YCOORD _ (IQUOTIENT (ITIMES (FONTPROP
                                                                                               MENUFONT
                                                                                               'HEIGHT)
                                                                                              (LENGTH (CDDR ITEM))
                                                                                              )
                                                                                      2]
         (SETQ EDITMENU\COMS COMS)
         (SETQ EDITMENU\SUBS SUBMENUS)
         (CHECK/MENU/IMAGE (create MENU
                                   ITEMS _ COMLIST
                                   CENTERFLG _ T])
```

(FINDEDITCOM

```
[LAMBDA (C L EFLG)                                          (* bas%: "19-NOV-82 15:28")
   (for I on L thereis (OR (EQUAL C (CAR (CADR I)))
                           (AND EFLG (NOT (CDR I])
```

(READEDITMENU

```
[LAMBDA NIL                                                 (* lmm " 4-Nov-85 22:47")
   (DECLARE (GLOBALVARS EDITMENU\COMS EDITMENU\SUBS))
   (bind OTHERS VAL N OLDN MOUSEISDOWN MOUSEWASDOWN EMDS (VLF _ (WINDOWPROP \DEDITMNUW 'ITEMHEIGHT))
      first (PROGN [SETQ \DEDITMNUW (SETQ EMDS (WINDOWPROP \DEDITMNUW 'DSP]
                                                            (* Clear menu to protect against ^E)
                   )
      eachtime (GETMOUSESTATE) while (AND (EQ \DEDITMNUW EMDS)
                                          (NOT (READP T))
                                          (OR (COND
                                                 ((SHIFTDOWNP 'CTRL)
                                                  (COND
                                                     (VAL (SHADEMENUENTRY N EMDS VLF 'HOLLOW OTHERS)
                                                          (push OTHERS (CONS N VAL))
                                                          (SETQ VAL NIL)))
                                                  OTHERS))
                                              (INWINDOW EMDS))
                                          (NOT VAL))
      when (INWINDOW EMDS) do (SETQ OLDN N)
                              (SETQ N (IQUOTIENT (LASTMOUSEY EMDS)
                                                 VLF))
                              [COND
                                 ((AND [EQ (SETQ MOUSEWASDOWN MOUSEISDOWN)
                                           (SETQ MOUSEISDOWN (LASTMOUSESTATE (NOT UP]
                                       (EQ N OLDN))         (* Nothing going on)
                                  (OR MOUSEISDOWN (BLOCK))  (* But dont block if mouse is down lest we miss an upclick)
                                  )
                                 (T (COND
                                       ((EQ N OLDN)
                                        (SHADEMENUENTRY N EMDS VLF 'HOLLOW OTHERS))
                                       (T (SHADEMENUENTRY OLDN EMDS VLF MOUSEWASDOWN OTHERS)
                                          (SHADEMENUENTRY N EMDS VLF MOUSEISDOWN OTHERS)))
                                    (COND
                                       ((AND (LASTMOUSESTATE MIDDLE)
                                             (ELT EDITMENU\SUBS N))
                                                            (* Submenu)
                                        (SETQ VAL (MENU (ELT EDITMENU\SUBS N)))
```

```
                                        (SETQ MOUSEISDOWN NIL)
                                        (SHADEMENUENTRY N EMDS VLF 'HOLLOW OTHERS))
                                       ((AND (NOT MOUSEISDOWN)
                                             MOUSEWASDOWN N)   (* Mouse has come up and a com is selected)
                                        (SETQ VAL (ELT EDITMENU\COMS N]
      finally (SHADEMENUENTRY N EMDS VLF MOUSEISDOWN OTHERS)
             (for I on OTHERS do (SHADEMENUENTRY (CAAR I)
                                        EMDS VLF 'FILL (CDR I)))
             [AND VAL OLDN (WINDOWPROP EMDS 'YOFFSET (ITIMES VLF (ADD1 OLDN]
             (SETQ \DEDITMNUW (COND
                                (\DEDITMNUW (WFROMDS EMDS))
                                (T EMDS)))                    (* Exited cleanly, restore global)
             (RETURN (COND
                        [OTHERS (AND VAL (bind CS XS for I in (CONS (CONS OLDN VAL)
                                                                   OTHERS)
                                           do (push CS (CADR I))
                                              (push XS (MKLIST (CDDR I)))
                                           finally (RETURN (CONS CS (CONS 'PROGN XS]
                        (T VAL])
```

(SHADEMENUENTRY

```
  [LAMBDA (V EMDS DLF BOXFLG OTHERS)                          ; Edited 11-Jun-90 14:53 by mitani
                                                              (* BOXFLG encoding%: T=FILL NIL=BOX for common cases of
    MOUSEDOWN controls)
    (AND V (NOT (FASSOC V OTHERS))
         (PROG [(D (SELECTQ BOXFLG
                        ((FILL T)
                            0)
                        (HOLLOW 1)
                        ((BOX NIL)
                            (SHADEMENUENTRY V EMDS DLF 'FILL)
                            1)
                        (SHOULDNT]
               (BITBLT NIL NIL NIL EMDS D (IPLUS D (ITIMES V DLF))
                      (IDIFFERENCE (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL EMDS))
                             (IPLUS D D))
                      (IDIFFERENCE DLF (IPLUS D D))
                      'TEXTURE
                      'INVERT BLACKSHADE])
```

(DEDITMENURESTORE

```
  [LAMBDA (W R)                                               (* bas%: " 5-Apr-84 19:56")
    (BITBLT (WINDOWPROP W 'IMAGE)
           1 1 W 0 0 NIL NIL 'INPUT 'REPLACE NIL R])

)

(RPAQQ *DEDIT-MENU-COMMANDS*
       [(After DEDITAfter)
        (Before DEDITBefore)
        (Delete DEDITDelete)
        (Replace DEDITReplace)
        (Switch DEDITSwitch)
        ("( )" DEDITBI ("( ) in" DEDITBI)
               ("( in" DEDITLI)
               (") in" DEDITRI))
        ("( ) out" DEDITBO ("( ) out" DEDITBO)
               ("( out" DEDITLO)
               (") out" DEDITRO))
        (Undo DEDITUndo (Undo DEDITUndo)
              (!Undo (DEDITUndo T))
              (?Undo (UNDOCHOOSE))
              (&Undo (UNDOCHOOSE T)))
        (Find DEDITFind)
        (Swap DEDITSwap (Center DEDITCenter)
              (Clear (SETQ \DEDITSELECTIONS NIL))
              (Copy DEDITCopy)
              (Pop (POPSELECTION))
              (Swap DEDITSwap))
        (Reprint DEDITReprint)
        [Edit DEDITEdit [DEdit (DEDITEdit 'DISPLAY 'Def)
                               NIL
                               (SUBITEMS ("DEdit Def" (DEDITEdit 'DISPLAY 'Def))
                                      ("DEdit Form" (DEDITEdit 'DISPLAY 'Form]
              [TTYEdit (DEDITEdit 'TELETYPE 'Def)
                     NIL
                     (SUBITEMS ("TTYEdit Def" (DEDITEdit 'TELETYPE 'Def))
                            ("TTYEdit Form" (DEDITEdit 'TELETYPE 'Form]
              (TTYIn% Form (DEDITEdit 'DEDIT.TTYinEdit 'Form]
        [EditCom DEDITEditCom (?= DEDITARGS)
               (GETD (DEDITEditCom 'GETD))
               (CL (DEDITEditCom 'CL))
               (DW (DEDITEditCom 'DW))
               (REPACK (DEDITEditCom 'REPACK))
               (CAP (DEDITEditCom 'CAP))
```

```
               (LOWER (DEDITEditCom 'LOWER))
               (RAISE (DEDITEditCom 'RAISE]
         (Break DEDITBreak)
         (Eval DEDITEval)
         (Exit DEDITExit (OK DEDITExit)
               (STOP (DEDITExit T])

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS *DEDIT-MENU-COMMANDS*)
)

;; Maintaining deditmap entries and the edit chain

(DEFINEQ

(BUFSELP
  [LAMBDA (E)                                                    (* bas%: "21-MAR-83 19:53")
    (AND E \DEDITBUFW (EQ (fetch PDSP of E)
                          (WINDOWPROP \DEDITBUFW 'DSP])

(EDITWINDOWP
  [LAMBDA (W)                                                    (* rmk%: " 1-SEP-83 11:23")
    (AND (OR (WINDOWP W)
             (DISPLAYSTREAMP W))
         (WINDOWPROP W 'EDITEXPR)
         (WINDOWPROP W 'DSP])

(GETLEFT
  [LAMBDA (SEL BK)                                               (* bas%: "16-MAR-83 09:45")
    (AND (OR BK (SETQ BK (fetch BP of SEL)))
         (for I on (fetch SELEXP of BK) when (COND
                                               ((LISTP (CDR I))
                                                (EQ (CDR I)
                                                    (fetch TAIL of SEL)))
                                               ((CDR I)
                                                (EQ (CDR I)
                                                    (fetch SELEXP of SEL)))
                                               (T NIL))
            do (RETURN (GETME4 I])

(GETMEBP
  [LAMBDA (E)                                                    (* bas%: "13-OCT-81 16:21")
    (OR (fetch BP of E)
        (CANT "At top"])

(HASASBP
  [LAMBDA (M F)                                                  (* bas%: "11-Mar-84 21:57")
    (OR (TAILP (OR (LISTP M)
                   (SETQ M (fetch TAIL of M)))
               (fetch SELEXP of F))
        (AND (NLISTP (CDR M))
             (EQ M (GETHASH (fetch TAIL of F)
                            \DEDITDPHASH])

(TAILOF
  [LAMBDA (A B)                                                  (* bas%: "11-Mar-84 23:31")
    (OR (TAILP A B)
        (AND (SETQ A (DPCDRSEL A))
             (SETQ B (GETME4 B))
             (EQ (fetch BP of A)
                 (fetch BP of B])

(DOTTEDEND
  [LAMBDA (C)                                                    (* bas%: "16-MAR-83 21:32")
    (COND
       ((LISTP C)
        (CDR (LAST C)))
       (T C])

(GETME4
  [LAMBDA (C B)                                                  (* bas%: "11-Mar-84 23:09")
    (AND C (OR (GETHASH C \DEDITMEHASH)
               (SELECTQ B
                    (NIL NIL)
                    (T (SHOULDNT "No MapEntry"))
                    (PROGN (OR (MAPENTRYP B)
                               (SETQ B (GETME4 B T)))
                           (OR [COND
                                  ((LISTP C)
```

```
                                (HASASBP C B))
                               (T (EQ C (DOTTEDEND (fetch SELEXP of B]
                             (SHOULDNT "Invalid BP")))
                          (COND
                             ((NLISTP C)
                              (GETDPME B))
                             [(MAPENTRYP (MAPHASH \DEDITMEHASH
                                         (FUNCTION (LAMBDA (X Y)
                                                   (AND (EQ B (fetch BP of X))
                                                        (EQUAL C Y)
                                                        (PROGN (PUTHASH Y NIL \DEDITMEHASH)
                                                               (replace TAIL of X with C)
                                                               (PUTHASH C X \DEDITMEHASH)
                                                               (RETFROM 'MAPHASH X]
                             (T (DEARME B])
```

(GETSELMAP

```
[LAMBDA (X)                                                    (* bas%: "12-Sep-84 10:40")
```

(* Gets ME iff it is unpurged and not a dummy ie visible for a SHADESELECTIOn etc)

```
  (AND (SETQ X (GETME4 X))
       (NEQ (fetch STARTX of X)
            (fetch STOPX of X))
       (UNPURGEDP X])
```

(DEARME

```
[LAMBDA (B)                                                    (* bas%: " 7-MAR-83 22:49")
  (REPP B)
  (for (SP _ (REALSTKNTH -1 'GETME4)) by (STKPOS (STKNAME SP)
                                                 -1
                                                 (STKNTH -1 SP SP)
                                                 SP)
     while SP when (EQ B (STKARG 1 SP)) do (RETEVAL SP [CONS (STKNAME SP)
                                                              (CONS (GETME4 (fetch TAIL of B)
                                                                            T)
                                                                    (CDR (STKARGS SP]
                                                       T)
     finally (RETURN (GETME4 (fetch TAIL of B)
                             T])
```

(DPCDRSEL

```
[LAMBDA (ME)                                                   (* bas%: "21-MAR-83 19:58")
  (AND [OR (type? DEDITMAP ME)
           (AND (CDR (LISTP ME))
                (NLISTP (CDR ME))
                (SETQ ME (GETME4 ME]
       (fetch BP of ME)
       (EQ ME (GETDPME (fetch BP of ME)))
       ME])
```

(GETDPME

```
[LAMBDA (B)                                                    (* bas%: "21-MAR-83 19:48")
  (GETME4 (GETHASH (fetch TAIL of B)
                   \DEDITDPHASH)
          T])
```

(GETEBUF

```
[LAMBDA (EW)                                                   (* bvm%: "27-May-86 15:15")
```

(* * "Return the edit buffer window for main window EW, reshaping or moving it as needed if windows have moved in the meantime. Maybe should do this with attached windows.")

```
  (LET ((MAINREG (WINDOWPROP EW 'REGION))
        (EBW (WINDOWPROP EW 'EDITBUF))
        EBWREG LEFT)
       (COND
          ((AND \DEDITBUFW (NEQ EBW \DEDITBUFW))
           (CLOSEW \DEDITBUFW)))
       (COND
          ((NOT EBW)
           (SETQ EBW (CREATEW (GETEBUFREGION MAINREG (OR (FIXP *DEDIT-BUFFER-HEIGHT*)
                                                         60)
                                                     EW)
                              "Edit buffer"))
           (WINDOWPROP EBW 'PAGEFULLFN (FUNCTION NILL))
           (WINDOWPROP EW 'EDITBUF EBW))
          ((PROGN (PURGEW (ACTIVEEDITW EBW NIL))
                  (SETQ EBWREG (WINDOWPROP EBW 'REGION))
                  (NEQ (fetch (REGION WIDTH) of MAINREG)
                       (fetch (REGION WIDTH) of EBWREG)))          (* "User reshaped edit window to different width. Reshape it now to the main window's width, user's height. No DEdit specific reshaping will happen because window is now inactive")
```

```
            (SHAPEW EBW (GETEBUFREGION MAINREG (fetch (REGION HEIGHT) of EBWREG)
                               EBW)))
           ((NEQ (SETQ LEFT (fetch (REGION LEFT) of MAINREG))
                 (fetch (REGION LEFT) of EBWREG))              (* "Window strayed somehow, move it to the right place")
            (MOVEW EBW LEFT (IDIFFERENCE (fetch (REGION BOTTOM) of MAINREG)
                                         (fetch (REGION HEIGHT) of EBWREG)))
            (OPENW EBW))
           (T (OPENW EBW)))
        (WINDOWPROP EBW 'READER-ENVIRONMENT (WINDOWPROP EW 'READER-ENVIRONMENT))
        (WINDOWPROP (SETQ \DEDITBUFW EBW)
               'DSP])
```

(GETEBUFREGION

```
[LAMBDA (MAINREG HEIGHT EW)                                   (* bvm%: "27-May-86 15:07")
  (LET* ((FONTHEIGHT (FONTPROP EW 'HEIGHT))
         (TOTALHEIGHT (HEIGHTIFWINDOW HEIGHT T))
         (BOTTOM (IDIFFERENCE (fetch (REGION BOTTOM) of MAINREG)
                              TOTALHEIGHT))
         EXCESS)
        [COND
           ((LESSP BOTTOM 0)                                  (* "Region overlaps bottom of screen, so force it on")
            (SETQ BOTTOM 0)
            [SETQ HEIGHT (IDIFFERENCE HEIGHT (IDIFFERENCE TOTALHEIGHT (SETQ TOTALHEIGHT
                                                                            (IDIFFERENCE (fetch (REGION BOTTOM)
                                                                                         of MAINREG)
                                                                                         BOTTOM]
           ((LESSP HEIGHT 0)                                  (* "Eek, it's off screen entirely.  Make it one high just for giggles")
            (SETQ TOTALHEIGHT (HEIGHTIFWINDOW (SETQ HEIGHT FONTHEIGHT)
                                              T]
        [COND
           ((NEQ (SETQ EXCESS (IREMAINDER HEIGHT FONTHEIGHT))
                 0)                                           (* Try to make window integral number of lines high)
            (SETQ TOTALHEIGHT (IDIFFERENCE TOTALHEIGHT EXCESS))
            (SETQ BOTTOM (IPLUS BOTTOM EXCESS]
        (create REGION
               LEFT _ (fetch (REGION LEFT) of MAINREG)
               BOTTOM _ BOTTOM
               WIDTH _ (fetch (REGION WIDTH) of MAINREG)
               HEIGHT _ TOTALHEIGHT])
```

(GETEDITCHAIN

```
[LAMBDA (E)                                                   (* bas%: "30-MAR-83 21:45")
  (DECLARE (USEDFREE LASTAIL))
  (COND
     ((LISTP E)
      (SETQ LASTAIL E)
      (SETQ E (OR (GETME4 E)
                  E)))
     ((type? DEDITMAP E)
      (SETQ LASTAIL (fetch TAIL of E)))
     (E (SHOULDNT)))
  (OR (LISTP E)
      (for (I _ E) by (fetch BP of I) while I collect (fetch SELEXP of I])
```

(GETDEDITMAP

```
[LAMBDA (DS)                                                  (* bas%: "11-Mar-84 23:15")
  (OR (GETMAP? DS)
      (SETDEDITMAP DS (COND
                         ((ZORCHEDWP DS)
                          (LIST (GETDEDITDEF4 DS)))
                         (T (WINDOWPROP DS 'EDITEXPR])
```

(GETMAP?

```
[LAMBDA (W)                                                   (* bas%: " 8-Mar-84 14:38")
  (GETSELMAP (WINDOWPROP W 'EDITEXPR])
```

(UNPURGEDP

```
[LAMBDA (M)                                                   (* bas%: "11-Mar-84 23:09")
```

(* This is unfortunately an expensive operation as some edit operations can cut a cons out of the structure being edited without that being obvious at the time it happens. The only way therefore to be sure that a ME really is valid is to chase its BPs all the way out to the top.)

```
  (AND (EQ M (GETME4 (fetch TAIL of M)))
       [OR (NOT (fetch BP of M))
           (AND (HASASBP M (fetch BP of M))
                (UNPURGEDP (fetch BP of M]
       M])
```

(SUBSELOF

```
  [LAMBDA (TOP SUB)                                          (* bas%: " 8-Mar-84 14:11")
    (for (S2 _ (GETSELMAP SUB)) by (fetch BP of S2) while S2 thereis (EQ TOP (fetch SELEXP of S2])
```

(SETDEDITMAP

```
  [LAMBDA (DW V)                                             (* bas%: "24-Jun-84 17:33")
    (PURGEW DW)                                              (* Remove EDITEXPR and reset window)
    [SETQ V (DEPRINTDEF (MKLIST V)
                   (DSPLEFTMARGIN NIL DW)
                   DEFAULTFONT
                   (WINDOWPROP DW 'DSP]
    (WINDOWPROP DW 'EDITEXPR (fetch TAIL of V))
    [WINDOWPROP DW 'EXTENT (create REGION
                                   LEFT _ 0
                                   BOTTOM _ (LOWPT V)
                                   WIDTH _ (WINDOWPROP DW 'WIDTH)
                                   HEIGHT _ (ADD1 (IDIFFERENCE (HIPT V)
                                                               (LOWPT V]
    V])
```

(TAKEDOWN

```
  [LAMBDA (WDS)                                              (* bas%: " 4-Apr-84 13:27")
    (COND
       (WDS (PURGEW WDS T)
            (CLOSEW WDS])

)

(RPAQ? *DEDIT-BUFFER-HEIGHT* 60)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS *DEDIT-BUFFER-HEIGHT*)
)

(DEFINEQ
```

(DEDITRESHAPEFN

```
  [LAMBDA (W X1 X2)                                          (* bas%: " 4-Apr-84 13:12")
    (AND (EDITWINDOWP W)
         (RESETFORM (CURSOR WAITINGCURSOR)
                (SETDEDITMAP W (WINDOWPROP W 'EDITEXPR))
                (FLIPSELSIN W (IPLUS (WYOFFSET NIL W)
                                     (WINDOWPROP W 'HEIGHT))
                       (WYOFFSET NIL W])
```

(DEDITREPAINTFN

```
  [LAMBDA (WDS R)                                            ; Edited 11-Jun-90 14:53 by mitani
    (PROG ((H (fetch (REGION PTOP) of R))
           (L (fetch (REGION BOTTOM) of R)))
          (REFRESHIF WDS H L)
          (FLIPSELSIN WDS H L])

)

(DEFINEQ
```

(RESETDEDIT

```
  [LAMBDA NIL                                                (* jow "16-Oct-86 11:24")
    (DECLARE (GLOBALVARS \DEDITCOMS))
    (pushnew MARKASCHANGEDFNS (FUNCTION DEDITMARKASCHANGED))
    (PROGN ;; DEdit wants to save these definitions.  Must take them from TTY/ now because EDITINTERFACE moved them.
           (MOVD 'TTY/EDITL 'NORMAL/EDITL)
           (MOVD 'TTY/EDITDATE 'NORMAL\EDITDATE)
           (EDITMODE 'DEDIT))
    (PROGN (for I in (CONS DEditWindow (LISTP \DEDITWINDOWS)) when (WINDOWP I) do (CLOSEW I))
           (SETQ DEditWindow NIL)                            (* Initialize DEDIT globals)
           (SETQ \DEDITWINDOWS NIL)
           (SETQ \DEDITALLOWSELS NIL)
           (SETQ \DEDITSELECTIONS NIL)
           (SETQ \DEDITBUFW NIL)
           (SETQ \DEDITMNUW NIL)
           (SETQ \DEDITMEHASH (HASHARRAY 255))
           (SETQ \DEDITDPHASH (HASHARRAY 255))
           (SETQ \DEDITFONTS NIL)
           (SETQ \DEDITDSPS (ARRAY 8))                       (* 8 is arbitrary)
           )
    T])
```

(DEDITDATE

```
  [LAMBDA (OLDATE INITLS)                                    (* bas%: " 5-FEB-83 19:36")
    (PROG1 (NORMAL\EDITDATE OLDATE INITLS)
        (PROG (ODM W)
```

```
              (AND (SETQ ODM (GETME4 (LISTP OLDATE)))
                   (SETQ ODM (fetch BP of ODM))
                   [ACTIVEWP (SETQ W (WFROMDS (fetch PDSP of ODM]
                   (GETMAP? W)
                   (REPP ODM))))])
```

(DEDITMARKASCHANGED

```
  [LAMBDA (NAME TYPE REASON)                                    (* lmm "29-Jul-85 21:11")
```

(* MARKASCHANGED is called after DEDITL exits. Hence a scan of the \DEDITWINDOWS chain finds all active DEDITs excluding the one just exited. The separate test on DEditWindow discriminates between exit from topmost DEDIT and other changes to the top level window)

```
    (ZORCHEDITW (COND
                   ((FINDEDITW NAME TYPE))
                   (T (AND (WINDOWP DEditWindow)
                           (SAMEEDITW DEditWindow NAME TYPE)
                           (NOT (CADR (WINDOWPROP DEditWindow 'DEDITCHANGES NIL)))
                           DEditWindow])

)

(DEFINEQ
```

(COPYCONS

```
  [LAMBDA (C)                                                   (* bas%: "22-FEB-82 14:20")
    (CONS (CAR C)
          (CDR C])
```

(COPYOUTCONS

```
  [LAMBDA (C1 C2)                                               (* bas%: "18-Mar-84 15:09")
                                                                (* Returns C1 with any instances of C2 COPYCONSed out)
    (COND
       ((NLISTP C1)
        C1)
       ((EQ C1 C2)
        (COPYCONS C1))
       (T (PROG ((CA (COPYOUTCONS (CAR C1)
                                  C2))
                 (CD (COPYOUTCONS (CDR C1)
                                  C2)))
                (RETURN (COND
                           ((AND (EQ CA (CAR C1))
                                 (EQ CD (CDR C1)))
                            C1)
                           (T (CONS CA CD])
```

(MAPENTRYP

```
  [LAMBDA (V)                                                   (* bas%: "21-MAR-83 19:58")
    (AND (type? DEDITMAP V)
         V])
```

(THELIST

```
  [LAMBDA (X)                                                   (* bas%: "21-JUL-82 18:11")
    (OR (LISTP X)
        (CANT "Not a list!"])

)

(DEFINEQ
```

(CANT

```
  [LAMBDA NMSGS                                                 (* hdj " 7-May-86 11:09")
                                                                (* Report error by flashing window)
    (DSPRESET PROMPTWINDOW)
    (printout PROMPTWINDOW T "Can't: ")
    (for I to NMSGS do (printout PROMPTWINDOW %, (ARG NMSGS I)))
    (FLASHWINDOW PROMPTWINDOW)
    (ERROR!])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD STACK (TOPELT NXTELT)
       (CREATE NIL))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS EDITBLOCKCALL MACRO (F (CONS (PACK* '\EDITBLOCK/ (CAR F))
```

```
                                        (CDR F))))

(PUTPROPS CONTROLCODE MACRO [(CHAR)
                                 (IDIFFERENCE (CHCON1 CHAR)
                                        (CONSTANT (IDIFFERENCE (CHARCODE A)
                                                               (CHARCODE ^A])

(PUTPROPS OVERLAP MACRO [OPENLAMBDA (H1 L1 H2 L2)
                            (NOT (OR (ILESSP H1 L2)
                                     (ILESSP H2 L1])

(PUTPROPS SHIFTSELECTKEYS MACRO [NIL (OR (SHIFTDOWNP 'SHIFT)
                                         (KEYDOWNP 'COPY])
)

(DECLARE%: EVAL@COMPILE

(RPAQQ LINETHICKNESS 2)

(RPAQQ PRIMSHADE 65535)

(RPAQQ SECSHADE 3598)

(RPAQ SWITCHSHADE (LOGXOR PRIMSHADE SECSHADE))

(RPAQQ READSHADE 23130)

(RPAQQ CHANGEDSHADE 8840)

(CONSTANTS (LINETHICKNESS 2)
       (PRIMSHADE 65535)
       (SECSHADE 3598)
       (SWITCHSHADE (LOGXOR PRIMSHADE SECSHADE))
       (READSHADE 23130)
       (CHANGEDSHADE 8840))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEditWindow \DEDITMNUW \DEDITBUFW \DEDITALLOWSELS \DEDITWINDOWS \DEDITSELECTIONS DT.EDITMACROS
       UPFINDFLG)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(SPECVARS ATM EDITCHANGES EDITHIST LASTAIL UNDOLST UNDOLST1)
)
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD (LOADCOMP)
       DEDITPP)
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(FILESLOAD DSPRINTDEF NEWPRINTDEF DEDITPP)

(AND (GETD 'RESETDEDIT)
     (RESETDEDIT))
)

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA CANT)
)

(PUTPROPS DEDIT COPYRIGHT ("Venue & Xerox Corporation" T 1982 1983 1984 1985 1986 1990 1991))
```

FUNCTION INDEX

VARIABLE INDEX

CONSTANT INDEX

MACRO INDEX

PROPERTY INDEX

RECORD INDEX