

File created: 23-Oct-91 16:39:18 {PELE:MV:ENVOS}<LISPCORE>SOURCES>WEDIT.;4

changes to: (FNS EDITGETD)

previous date: 25-Jun-90 14:33:27 {PELE:MV:ENVOS}<LISPCORE>SOURCES>WEDIT.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1990, 1991 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **WEDITCOMS**

```
[ (VARS EDITOPS MAXLOOP (EDITRACEFN)
  (UPFINDFLG T)
  MAXLEVEL FINDFLAG (EDITQUIETFLG))
[ INITVARS (EDITSMASHUSERFN)
  (EDITEMBEDTOKEN '&)
  (EDITUSERFN)
  (CHANGESARRAY)
  (EDITUNSAVEBLOCKFLG T)
  (EDITLOADFNSFLG '(T))
[ INITVARS (EDITMACROS)
  (USERMACROS))
[ ADDVARS (HISTORYCOMS ?? REDO REPEAT FIX USE |...| NAME RETRIEVE DO !N !E !F TYPE-AHEAD Ÿ ŸBUFS ;)
  (DONTSAVEHISTORYCOMS SAVE P ? PP PP* E ;)
  (EDITCOMSA OK STOP SAVE TTY%: E ? PP PP* PPV P ^ !0 MARK UNDO !UNDO TEST UNBLOCK _ \ \P _ F BF
    UP DELETE NX BK !NX ?? REDO REPEAT FIX USE NAME RETRIEVE DO !N !E !F TYPE-AHEAD)
  (EDITCOMSL S R R1 RC RC1 E I N P F FS F= ORF BF NTH IF RI RO LI LO BI BO M NX BK ORR MBD XTR THRU
    TO A B %: AFTER BEFORE MV LP LPQ LC LCL _ BELOW SW BIND COMS ORIGINAL INSERT REPLACE
    CHANGE DELETE EMBED SURROUND MOVE EXTRACT SWITCH ?? REDO REPEAT FIX USE NAME RETRIEVE DO
    MARK \))
[ USERMACROS CAP LOWER RAISE 2ND 3RD %F %% NEX REPACK * >* SHOW EXAM PP* ?=)
[ COMS
  (FNS SETTERMCHARS INTCHECK CHARMACRO)
  (INITVARS (EDITCHARACTERS))
  (VARS NEGATIONS)
  (USERMACROS 2P NXP BKP -1P)
  (ADDVARS (COMPACTHISTORYCOMS 2P NXP BKP -1P))
  (DECLARE%: DONTCOPY (MACROS CFOBF))
  (BLOCKS (SETTERMCHARS SETTERMCHARS INTCHECK (NOLINKFNS . T)
    (GLOBALVARS EDITRDTBL))
    (NIL CHARMACRO (LOCALVARS . T))
[ COMS
  (USERMACROS EF EV EP)
  (ADDVARS (DONTSAVEHISTORYCOMS EF EV EP))
  (FNS FIRSTATOM)
  (BLOCKS (NIL FIRSTATOM (LOCALVARS . T))
    ; macros for calling editor
[ COMS
  (USERMACROS EVAL Q GETD GETVAL MAKEFN D NEGATE GO SWAP MAKE SWAPC IFY SPLITC JOINC)
  (FNS MAKEFN EDITGETD EDITGETD-LAMBDA EDITGETD-LET NEGATE NEGL NEGLST NEGC MKPROGN MKPROGN1 MAKECOM
    SWAPPEDCOND COND.TO.IF)
  (PROP EDITGETD LET CL:IF)
  (BLOCKS (NEGATE NEGATE NEGL NEGLST NEGC (NOLINKFNS . T)
    (GLOBALVARS NEGATIONS))
    (MKPROGN MKPROGN MKPROGN1 (NOLINKFNS . T))
  (GLOBALVARS CLISPARRAY MACROPROPS)
  (LOCALVARS . T)
  (GLOBALVARS LAMBDA$PLST NORMALCOMMENTSFLG COMMENTFLG FIRSTNAME INITIALS INITIALSLST DEFAULTINITIALS)
  (P (MOVD? 'NIL 'PREEDITFN))
  (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILE$VARS (ADDVARS (NLAMA)
    (NLAML MAKECOM CHARMACRO)
    (LAMA]))
```

(RPAQQ **EDITOPS**

```
((INSERT (BEFORE AFTER FOR)
  (EDIT%: %%2 %%3 %%1))
(REPLACE (WITH BY)
  (EDIT%: %: %%1 %%3))
(CHANGE (TO)
  (EDIT%: %: %%1 %%3))
(DELETE NIL (EDIT%: %: %%1))
(EMBED (IN WITH)
  (EDITMBD %%1 %%3))
(SURROUND (WITH IN)
  (EDITMBD %%1 %%3))
(MOVE (TO)
  (EDITMV %%1 (CAR %%3)
    (CDR %%3)))
(EXTRACT (FROM)
  (EDITXTR %%3 %%1))
(SWITCH (AND)
  (EDITSW %%1 %%3)))
```

```

(RPAQQ MAXLOOP 30)
(RPAQQ EDITRACEFN NIL)
(RPAQQ UPFINDFLG T)
(RPAQQ MAXLEVEL 300)
(RPAQQ FINDFLAG NIL)
(RPAQQ EDITQUIETFLG NIL)
(RPAQ? EDITSMASHUSERFN )
(RPAQ? EDITEMBEDTOKEN '&')
(RPAQ? EDITUSERFN )
(RPAQ? CHANGESARRAY )
(RPAQ? EDITUNSAVEBLOCKFLG T)
(RPAQ? EDITLOADFNSFLG '(T)')
(RPAQ? EDITMACROS )
(RPAQ? USERMACROS )

(ADDTOVAR HISTORYCOMS ?? REDO REPEAT FIX USE |...| NAME RETRIEVE DO !N !E !F TYPE-AHEAD Ÿ ŸBUFS ;)
(ADDTOVAR DONTSAVEHISTORYCOMS SAVE P ? PP PP* E ;)
(ADDTOVAR EDITCOMSA OK STOP SAVE TTY%: E ? PP PP* PPV P ^ !0 MARK UNDO !UNDO TEST UNBLOCK _ \ \P ___ F BF UP
DELETE NX BK !NX ?? REDO REPEAT FIX USE NAME RETRIEVE DO !N !E !F TYPE-AHEAD)
(ADDTOVAR EDITCOMSL S R R1 RC RC1 E I N P F FS F= ORF BF NTH IF RI RO LI LO BI BO M NX BK ORR MBD XTR THRU TO A
B %: AFTER BEFORE MV LP LPQ LC LCL _ BELOW SW BIND COMS ORIGINAL INSERT REPLACE CHANGE
DELETE EMBED SURROUND MOVE EXTRACT SWITCH ?? REDO REPEAT FIX USE NAME RETRIEVE DO MARK \)

(ADDTOVAR EDITMACROS
  (EXAM X (F (*ANY* . X)
    T)
    (BIND (LPQ (MARK %1)
      (ORR (1 !0 P)
        NIL)
      (MARK %2)
      TTY%:
      (MARK %3)
      (IF (EQ (### (\ %3))
        (### (\ %2)))
        ((\ %1))
        NIL)
      (F (*ANY* . X)
        N)))
    (E 'done))
  (>* (X)
    (BIND (MARK %1)
      0
      (_ ((*ANY* PROG PROGN COND SELECTQ LAMBDA NLAMBDA ASSEMBLE)
        --))
      (MARK %2)
      (E (SETQ %3 (SELECTQ (%## 1)
        ((COND (SELECTQ)
          2)
          1))
        T)
        (\ %1)
        (ORR (1 1)
          (1)
          NIL)
        (BELOW (\ %2)
          %3)
        (IF 'X [(ORR (NX (B X))
          ((IF (EQ (%## (\ %2)
            0 1)
            'PROG)
            NIL
            (BK))
            (A X))
            ((\ %2)
            (>* X]
            NIL)))
        (LOWER NIL UP (I 1 (L-CASE (%## 1)))
          1)
        (CAP NIL UP (I 1 (L-CASE (%## 1)
          T))

```

```

1)
[REPACK NIL (IF (LISTP (%##))
  (1)
  NIL)
  (I %: ([LAMBDA (X Y)
    (SETQ COM 'REPACK)
    [SETQ Y (APPLY 'CONCAT (EDITE (UNPACK X)
    [COND ((NOT (STRINGP X))
      (SETQ Y (MKATOM Y)
      (PRINT Y T T]
    (%##]
  (* X MARK [ORR [(I >* (COND [(RAISEP)
    (CONS '* (CONS '% 'X)
    (T (CONS '* 'X)
    ((E 'CAN'T]
  __)
[LOWER (C)
  (I R 'C (L-CASE 'C)
[RAISE (C)
  (I R (L-CASE 'C)
  'C))
[RAISE NIL UP (I 1 (U-CASE (%## 1)))
  1)
[2ND X (ORR ((LC . X)
  (LC . X)
[3RD X (ORR ((LC . X)
  (LC . X)
  (LC . X)
(%F (X Y)
  (E (EDITQF (L-CASE 'X 'Y))
  T))
[%% X (COMS (CONS (CAR 'X)
  (COMMENT3 (CDR 'X)
  (CAR (LAST L]
(NEX NIL (BELOW __)
  NX)
(NEX (X)
  (BELOW X)
  NX)
[REPACK NIL (IF (LISTP (%##))
  (1)
  NIL)
  (I %: ([LAMBDA (X Y)
    (SETQ COM 'REPACK)
    [SETQ Y (APPLY 'CONCAT (EDITE (UNPACK X)
    [COND ((NOT (STRINGP X))
      (SETQ Y (MKATOM Y)
      (PRINT Y T T]
    (%##]
  (REPACK (X)
    (LC . X)
  REPACK)
  (SHOW X (F (*ANY* . X)
    T)
    (LPQ MARK (ORR (1 !0)
      NIL)
      P __ (F (*ANY* . X)
        N))
    (E 'done))
  (PP* NIL (RESETVAR **COMMENT**FLG NIL PP))
  (?= NIL (E (PROGN (PRINT-ARGLIST (SMARTARGLIST (CAR (%##)))
    (CDR (%##))
    T 0)
    (TERPRI T))
  T)))

```

(ADDTOVAR **EDITCOMSA** CAP LOWER RAISE NEX REPACK PP*)

(ADDTOVAR **EDITCOMSL** LOWER RAISE 2ND 3RD %%F %% REPACK * >* EXAM SHOW)

(ADDTOVAR **DONTSAVEHISTORYCOMS** PP*)

:: control chars for moving around in the editor

(DEFINEQ

(SETTERMCHARS

```

[LAMBDA (NEXTCHAR BKCHAR LASTCHAR UNQUOTECHAR 2CHAR PPCHAR) (* lmm "11-SEP-78 04:57")
  (COND
    ((SETQ NEXTCHAR (INTCHECK NEXTCHAR)) ; NEXTCHAR (usu. control-J) goes to the next entry
    (/SETSNTAX NEXTCHAR '[MACRO FIRST IMMEDIATE (LAMBDA NIL
      (CHARMACRO NXP]
      EDITRDTBL)))
  [COND
    ((SETQ LASTCHAR (INTCHECK LASTCHAR)) ; LASTCHAR (usu. control-Z) to the editor will go to the last thing
    (/SETSNTAX LASTCHAR '[MACRO FIRST IMMEDIATE (LAMBDA NIL
      and print it

```

(CHARMACRO -1P]

```

      EDITRDTBL)
      (/ECHOCONTROL LASTCHAR 'IGNORE]
[COND
  ((SETQ 2CHAR (INTCHECK 2CHAR))
    ; 2CHAR (usu. Control N) to the editor will go to 2 (or 1) and print
    ; it
    (/SETSNTAX 2CHAR '[MACRO FIRST IMMEDIATE (LAMBDA NIL
      (CHARMACRO 2P]
      EDITRDTBL)
      (/ECHOCONTROL 2CHAR 'IGNORE]
[COND
  ((SETQ BKCHAR (INTCHECK BKCHAR))
    ; BKCHAR (usu. control H) to the editor will go back (or up) and
    ; then print
    (/SETSNTAX BKCHAR '[MACRO FIRST IMMEDIATE ESCQUOTE (LAMBDA NIL
      (CHARMACRO BKP]
      EDITRDTBL)
      (/ECHOCONTROL BKCHAR 'IGNORE]
[COND
  ((SETQ UNQUOTECHAR (INTCHECK UNQUOTECHAR))
    ; UNQUOTECHAR (usu. control Y (Yank)) is an 'unquote' --
    ; reads next thing and evals it
    (/SETSNTAX UNQUOTECHAR '[MACRO FIRST (LAMBDA (FILE RDTBL)
      (EVAL (READ FILE RDTBL)
        T)
      (/SETSNTAX UNQUOTECHAR T EDITRDTBL)))
[COND
  ((SETQ PPCHAR (INTCHECK PPCHAR))
    ; PPCHAR (usu. control-O) to the editor will print current
    ; expression
    (/SETSNTAX PPCHAR '[SPLICE FIRST IMMEDIATE ESCQUOTE (LAMBDA NIL
      (TERPRI T)
      (%## PP*)
      (PRIN1 '* T)
      NIL]
      EDITRDTBL)
      (/ECHOCONTROL PPCHAR 'IGNORE])

```

(INTCHECK

(* lmm "29-NOV-77 20:32")

```

[LAMBDA (CHAR)
  (PROG ((CHR CHAR)
    (NCHR)
    [COND
      ((LISTP CHR)
        (SETQ CHR (CAR CHR)
          (COND
            ((NULL CHR)
              (RETURN)))
            ((NOT (FIXP CHR))
              (SETQ CHR (CHCON1 CHR)
                (COND
                  ((IGREATERP CHR 64)
                    (SETQ CHR (IDIFFERENCE CHR 64)
                      (COND
                        ((NOT (GETINTERRUPT CHR))
                          (RETURN CHR)))
                        ((NLISTP CHR)
                          (PRIN1 "control-" T)
                          (PRIN1 (FCHARACTER (IPLUS CHR 64))
                            T)
                          (PRIN1 " is an interrupt and can't be an edit control-character" T)
                          (TERPRI T))
                        (T (COND
                          [(SETQ NCHR (CADR CHR))
                            (OR (FIXP NCHR)
                              (SETQ NCHR (CHCON1 NCHR)))
                            [COND
                              ((IGREATERP NCHR 64)
                                (SETQ NCHR (IDIFFERENCE NCHR 64)
                                  (INTCHAR (CONS NCHR (CDR (OR (INTCHAR CHR)
                                                                    (HELP)
                                                                    (T (INTCHAR CHR)))
                                                                    (RETURN CHR)))

```

(CHARMACRO

(* NOBIND "18-JUL-78 22:15")
; clear file output buffer; no-op on dorado

```

[LAMBDA (X)
  (CFOB)
  (TERPRI T)
  X])

```

(RPAQ? EDITCHARACTERS)

(RPAQQ NEGATIONS ((NEQ . EQ)
(NLISTP . LISTP)

(GO . GO)
(ERROR . ERROR)
(ERRORX . ERRORX)
(RETURN . RETURN)
(RETFROM . RETFROM)
(RETTO . RETTO)
(IGREATERP . ILEQ)
(ILESSP . IGEQ))

(ADDTOVAR EDITMACROS
(NXP NIL [ORR (NX)
(!NX (E (PRIN1 "> " T)
T))
(E (PROGN (SETQQ COM NX)
(ERROR!])
P)
[-1P NIL (ORR (-1 P)
(E (PROGN (SETQQ COM -1)
(ERROR!])
BKP NIL [ORR (BK)
(!0)
(E (PROGN (SETQQ COM BK)
(ERROR!])
P)
(2P NIL [ORR (2)
(1)
(E (PROGN (SETQQ COM 2)
(ERROR!])
P))

(ADDTOVAR EDITCOMSA NXP -1P BKP 2P)
(ADDTOVAR COMPACTHISTORYCOMS 2P NXP BKP -1P)
(ADDTOVAR COMPACTHISTORYCOMS 2P NXP BKP -1P)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PROGN (PUTPROPS CFOBFB MACRO (NIL (ASSEMBLE NIL (MOVEI 1 %, 65)
(JSYS 65))))
(PUTPROPS CFOBFB DMACRO (NIL NIL)))
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: SETTERMCHARS SETTERMCHARS INTCHECK (NOLINKFNS . T)
(GLOBALVARS EDITRDTBL))

(BLOCK%: NIL CHARMACRO (LOCALVARS . T))
)

:: macros for calling editor

(ADDTOVAR EDITMACROS
[EV NIL (ORR [(E (LISPXEVAL (LIST 'EDITV (FIRSTATOM (%##)))
'EV->]
((E 'EV?])
[EP NIL (ORR [(E (LISPXEVAL (LIST 'EDITP (FIRSTATOM (%##)))
'EP->]
((E 'EP?])
[EF NIL (ORR [(E (LISPXEVAL (LIST 'EDITF (FIRSTATOM (%##)))
'EF->]
((E 'EF?])

(ADDTOVAR EDITCOMSA EV EP EF)

(ADDTOVAR DONTSAVEHISTORYCOMS EF EV EP)

(ADDTOVAR DONTSAVEHISTORYCOMS EF EV EP)

(DEFINEQ

(FIRSTATOM

[LAMBDA (X)

(* NOBIND "18-JUL-78 21:57")
; Used by EF macro

(COND

((NLISTP X)

X)

(T (FIRSTATOM (CAR X)))

)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

```
(BLOCK%: NIL FIRSTATOM (LOCALVARS . T))
)
```

;; Misc edit macros

```
(ADDTOVAR USERMACROS (IFY NIL (F (COND --)
                                     T)
                              UP
                              [I 1 (COND.TO.IF (CDR (%## 1)
                                                  1))
                              ]))
```

```
(ADDTOVAR EDITMACROS
  (SWAP (LC1 LC2)
    (BIND (MARK %#3)
           (LC . LC1)
           UP
           (MARK %#1)
           (\ %#3)
           (LC . LC2)
           UP
           [IF (NOT (OR (FMEMB (CAAR %#1)
                              L)
                        (FMEMB (CAAR L)
                              %#1)))
              ((E (SETQ %#2 (CAR (%##)))
                 T)
               (\ %#1)
               (E (SETQ %#1 (CAR (%##)))
                 T)
               (I 1 %#2)
               \
               (I 1 %#1))
              ((E '(NESTED EXPRESSIONS)
                 (\ %#3)))
              ]))
  [EVAL NIL (ORR [(E (LISPXEVAL (%## (ORR (UP 1)
                                          NIL))
                                '*]
                (E 'EVAL?)
                ])]
  [GO (LAB)
    (ORR (( _ ((*ANY* PROG ASSEMBLE DPROG RESETLST)
              -- LAB --))
          F LAB (ORR 2 1)
          P)
    (E (PROGN (SETQQ COM LAB)
              (ERROR!)]
  (JOINC NIL (F COND T)
    UP
    (BI 1 2)
    1
    (BO 2)
    (2)
    (RO 1)
    (BO 1))
  (NEGATE NIL UP (I 1 (NEGATE (%## 1)))
    1)
  (SPLITC (X)
    (F COND T)
    (BI 1 X)
    (IF [AND (EQ (%## 2 1)
                 T)
         (%## 2 2)
         (NULL (CDDR (%##)
                     ((BO 2)
                      (2)
                      ((-2 COND)
                       (LI 2))))
          ])]
    UP
    (BO 1))
  (SWAPC NIL (F ((*ANY* COND IF if)
                --)
            T)
    UP
    (I 1 (SWAPPEDCOND (%## 1)))
    1)
  (MAKE (VAR . VALS)
    (COMS (MAKECOM VAR VALS)))
  (D NIL (%:)
    1 P)
  (Q NIL (MBD QUOTE))
  (MAKEFN (FORM ARGS N M)
    [IF 'M ((BI N M)
            (LC . N)
            (BELOW \))]
    ((IF 'N ((BI N -1)
            (LC . N)
            (BELOW \)))
    ((LI 1]
```

```

(E (MAKEFN 'FORM 'ARGS (%##)
  T)
  UP
  (1 FORM)
  1)
(GETD NIL UP [ORR [(I 1 (OR [EDITGETD (%## 1)
                          (AND (CDR L)
                               (EDITLO L '(!0]
                          (ERROR!])
                          ((E 'GETD?])
                          1)
(GETVAL NIL UP [ORR [(I 1 (EVAL (%## 1)
                              ,*)]
                    ((E 'GETVAL?])
                    1))

```

(ADDTOVAR **EDITCOMSA** JOINC EVAL NEGATE SWAPC D Q GETD GETVAL)

(ADDTOVAR **EDITCOMSL** SPLITC MAKE MAKEFN SWAP GO)

(DEFINEQ

MAKEFN

[LAMBDA (FORM ARGLIST BODY)

(* Imm "10-Jun-85 01:37")
; called from MAKEFN edit macro

```

(COND
  ((AND (LITATOM FORM)
        (FNTYP (CAR BODY))
        (NULL (CDR BODY))
        (NULL ARGLIST))
    (DEFINE (LIST (CONS FORM BODY))
      T))
  (T (PROG ((A ARGLIST)
            (ACTUAL (CDR FORM))
            DEF)
        (OR (AND (LISTP FORM)
                 (CAR FORM)
                 (LITATOM (CAR FORM)))
            (ERROR FORM "? " T))
        LP (COND
            ((LISTP ACTUAL)
             [COND
              ((NLISTP A)
               (SETQ ARGLIST
                    (NCONC ARGLIST
                          (SETQ A
                               (LIST (COND
                                     ((LITATOM (CAR ACTUAL))
                                     (CAR ACTUAL))
                                     (T (OR [CAR (SOME '(X Y Z A B C D)
                                                       (FUNCTION (LAMBDA (X)
                                                                (NOT (FMEMB X ARGLIST])
                                                                (GENSYM)
                                                                (AND (NEQ (CAR A)
                                                                (CAR ACTUAL))
                                                                (ERSETQ (ESUBST (CAR A)
                                                                (CAR ACTUAL)
                                                                BODY)))
                                                                (SETQ A (CDR A))
                                                                (SETQ ACTUAL (CDR ACTUAL))
                                                                (GO LP)))
                                                                (DEFINE [LIST (LIST (CAR FORM)
                                                                (SETQ DEF (CONS 'LAMBDA (CONS ARGLIST BODY]
                                                                T])

```

EDITGETD

[LAMBDA (X EDITCHAIN)

; Edited 23-Oct-91 16:34 by jds

;;; used by the GETD edit macro and by the SEdit command Meta-X.

;; (CAR X) isn't guaranteed to be either a list or a litatom, so we have to check carefully.

```

(AND (LISTP X)
  (LET (DEF (FN (CAR X)))
    (COND
      ((LISTP FN)
       (EDITGETD-LAMBDA (CADR FN)
                          (CDDR FN)
                          (CDR X)))
      ([SETQ DEF (AND (LITATOM FN)
                     (GET FN 'EDITGETD]
       (COPY (CL:FUNCALL DEF X)))
      ((AND (LITATOM FN)
            (GET FN 'MACRO-FN))
       (COPY (CL:MACROEXPAND-1 X)))
      ((AND (SETQ DEF (EXPANDMACRO X T))
            (NEQ DEF X))

```

; a *REAL* common lisp macro, not a fake one

; This will catch Interlisp macros that are invisible to the new
; compiler but visible to the old one.

```

(COPY DEF))
((AND (LITATOM FN)
      (GETPROP FN 'CLISPWORD))
 [DWIMIFY X T (OR EDITCHAIN ' (NIL]
(COND
  ((NEQ FN (CAR X)) ; form changed
   X)
  ((SETQ DEF (GETHASH X CLISPARRAY))
   (COPY DEF))
  (T X)))
((AND (SETQ DEF (GETDEF FN 'FUNCTIONS NIL 'NOERROR))
      (EQ (CAR DEF)
          'CL:DEFUN))
 (EDITGETD-LAMBDA (CADDR DEF)
                  (CDDR DEF)
                  (CDR X)))
[(SETQ DEF (GETDEF FN 'FNS NIL 'NOERROR))
 (COND
  [(EQ (CAR DEF)
        'NLAMBDA)
   (EDITGETD-LAMBDA (CADR DEF)
                     (CDDR DEF)
                     (MAPCAR (CDR X)
                              (FUNCTION KWOTE]
                     (AND (EQ (CAR DEF)
                              'LAMBDA)
                          (CADR DEF)
                          (NLISTP (CADR DEF)))) ; LAMBDA spread
                     (CONS DEF (CDR X)))
   (T (EDITGETD-LAMBDA (CADR DEF)
                        (CDDR DEF)
                        (CDR X]
      (T X])

```

(EDITGETD-LAMBDA

(* Imm " 5-Jun-86 01:41")

```

[LAMBDA (ARGS BODY VALS DONE)
 (if (NULL ARGS)
     then (if DONE
            then ` (LET , (REVERSE DONE)
                    ,@BODY)
            else (MKPROGN BODY))
     elseif (CL:CONSP (CAR ARGS))
     then (EDITGETD-LAMBDA (CDR ARGS)
                            BODY
                            (CDR VALS)
                            (CONS (LIST (CAAR ARGS)
                                         (if VALS
                                             then (CAR VALS)
                                             else (CADAR ARGS)))
                                DONE))
     else (SELECTQ (CAR ARGS)
                  ((&ALLOW-OTHER-KEYS &OPTIONAL)
                   (EDITGETD-LAMBDA (CDR ARGS)
                                       BODY VALS DONE))
                  (&AUX (EDITGETD-LAMBDA ARGS (APPEND VALS BODY)
                          NIL DONE))
                  ((&BODY &REST)
                   [EDITGETD-LAMBDA (CDDR ARGS)
                                       BODY NIL `((, (CADR ARGS)
                                         (LIST ,@VALS))
                                         ,@DONE])
                   (EDITGETD-LAMBDA (CDR ARGS)
                                       BODY
                                       (CDR VALS)
                                       (CONS (LIST (CAR ARGS)
                                                  (CAR VALS))
                                             DONE])

```

(EDITGETD-LET

(* Imm " 5-Jun-86 01:44")

```

[LAMBDA (FORM)
 (MKPROGN (SUBPAIR (MAPCAR (CADR FORM)
                          (FUNCTION CAR))
              (MAPCAR (CADR FORM)
                      (FUNCTION CADR))
              (CDDR FORM])

```

(NEGATE

(* JonL "10-Apr-84 22:05")

```

[LAMBDA (X)
 (SELECTQ (CAR (LISTP X))
  ((NOT NULL)
   (CADR X))
  (AND (CONS 'OR (NEGLST (CDR X))))
  (OR (CONS 'AND (NEGLST (CDR X))))
  (COND [COND

```



```

[[[AND (NULL (CDDR X))
      (NULL (CDR (CDADR X))
      (NEGATE (CONS 'AND (CADR X))
      (T (CONS 'COND (NEGC (CDR X))
      (SELECTQ [CONS 'SELECTQ (CONS (CADR X)
      (MAPLIST (CDDR X)
      (FUNCTION (LAMBDA (X)
      (COND
      [(CDR X)
      (CONS (CAAR X)
      (NEGL (CDAR X))
      (T (NEGATE (CAR X))

      (PROGN (MKPROGN (NEGL (CDR X))))
      (PROG1 (CONS 'PROG1 (CONS (NEGATE (CADR X))
      (CDDR X))))
      (QUOTE (NULL (CADR X)))
      ((CONS
      (MKPROGN (APPEND (CDR X)
      (LIST NIL))))
      (COND
      [(for Y in NEGATIONS do (COND
      [(EQ (CAR Y)
      (CAR X))
      (RETURN (CONS (CDR Y)
      (CDR X))
      ((EQ (CDR Y)
      (CAR X))
      (RETURN (CONS (CAR Y)
      (CDR X))

      (T (OR (NULL X)
      (AND (NEQ X T)
      (NOT (OR (NUMBERP X)
      (STRINGP X)))
      (LIST 'NOT X])

```

; functions which always return non-NIL

(NEGL

```

[LAMBDA (L)
  (COND
  [(NULL (CDR L))
  (LIST (NEGATE (CAR L))
  (T (CONS (CAR L)
  (NEGL (CDR L))

```

(* Imm%: "7-FEB-77 17:17:51")

(NEGLST

```

[LAMBDA (L)
  (MAPCAR L (FUNCTION NEGATE))

```

(NEGC

```

[LAMBDA (X)
  (COND
  ((NULL X)
  (LIST (LIST T T)))
  [(NULL (CDAR X))

  (LIST (LIST (NEGATE (CAAR X))
  (OR (NULL (CDR X))
  (AND (SETQ X (NEGC (CDR X)))
  (CONS 'COND X))
  (T (CONS (CONS (CAAR X)
  (NEGL (CDAR X)))
  (AND (NEQ (CAAR X)
  T)
  (NEGC (CDR X))

```

(* Imm "14-SEP-78 23:07")

(* (COND (A) . TAIL) -> (NOT (OR A (COND . TAIL))) -> (AND (NOT A) (NOT (COND . TAIL))))

(MKPROGN

```

[LAMBDA (L)
  (COND
  ((CDR (SETQ L (MKPROGN1 L)))
  (CONS 'PROGN L))
  (T (CAR L))

```

(* wt%: "18-JUL-78 12:57")

(MKPROGN1

```

[LAMBDA (L)
  (COND
  ((NULL (CDR L))
  (COND
  ((EQ (CAAR L)
  'PROGN)
  (CDAR L))
  (T L)))
  ((NLISTP (CAR L))

```

(* Imm "21-SEP-77 15:19")

```

(MKPROGN1 (CDR L)))
(T (SELECTQ (CAAR L)
  ((PROGN LIST CONS CAR CDR NOT NULL)
   (MKPROGN1 (APPEND (CDAR L)
                     (CDR L))))
  (QUOTE (MKPROGN1 (CDR L)))
  (CONS (CAR L)
        (MKPROGN1 (CDR L]))

```

(MAKECOM

```

[NLAMBDA (VAR VALS) (* wt%: "19-JUL-78 11:35")
  (PROG (ARGNAMES (FORM (%##)))
    (SETQ ARGNAMES (SMARTARGLIST (SETQ COM (CAR FORM))
                                 NIL FORM))
    (OR [AND (LISTP ARGNAMES)
          (OR (FMEMB (SETQ COM VAR)
                    ARGNAMES)
              (SETQ VAR (FIXSPELL VAR NIL (APPEND ARGNAMES)
                                NIL]
        (ERROR!))
    (RETURN (PROG (($LST2 ARGNAMES)
                  $$VAL I ARG LST) (* (FOR I FROM 2 AS ARG IN VALS UNTIL ARG=VAR DO --))
            (SETQ I 2)
            $$LP
            [SETQ ARG (CAR (OR (LISTP $$LST2)
                              (GO $$OUT)
            [COND
              ((EQ ARG VAR)
               (RETURN (COND
                        ((NOT (OR VALS (NULL (CDR FORM))
                                         (CDDR FORM)))
                          (LIST I))
                        [(CDR FORM)
                         (LIST I (COND
                                ((CDR VALS)
                                 VALS)
                                (T (CAR VALS]
                                (T (CONS 'N (NCONC1 LST (COND
                                                ((CDR VALS)
                                                 VALS)
                                                (T (CAR VALS]
              [COND
                ((NULL (SETQ FORM (CDR FORM)))
                 (SETQ LST (CONS NIL LST]
            $$ITERATE
            (SETQ I (IPLUS I 1))
            (SETQ $$LST2 (CDR $$LST2))
            (GO $$LP)
            $$OUT
            (ERROR!)
            (RETURN $$VAL]))

```

(SWAPPEDCOND

```

[LAMBDA (CND) (* lmm "25-SEP-83 23:50")
  (SELECTQ (CAR CND)
    ((IF if)
     (DWIMIFY CND T L)
     [COND.TO.IF (CDR (SWAPPEDCOND (COND
                                   ((EQ (CAR CND)
                                        'COND)
                                    CND)
                                   ((GETHASH CND CLISPARRAY))
                                   (T (ERROR!]))
      (COND [PROG ((C1 (CADR CND))
                  (CTAIL (CDDR CND))
                  (C2 (CADDR CND)))
            (if (NOT (CDR C1))
                then ; cannot negate (COND (A) --)
                (ERROR!))
            (RETURN (CONS 'COND (CONS [CONS (NEGATE (CAR C1))
                                           (OR (AND (EQ (CAR C2)
                                                    T)
                                               (CDR C2))
                                           (COND
                                             ((NULL CTAIL)
                                              ; only one clause. Turn (COND (A --)) into (COND ((NOT A) NIL)
                                              ; (T --))
                                             (LIST NIL))
                                           (T ; embed multiple subsequent clauses into one COND
                                            (LIST (CONS 'COND CTAIL]
            (COND
              ((AND (NULL (CDDR C1))
                   (EQ (CAADR C1)
                       'COND))
               ; consequent of first clause is a COND itself. Expand out in the
               ; tail

```

```

(CDADR C1))
((AND (NULL (CADR C1))
      (NULL (CDDR C1)))
      ;(COND (A NIL --) when swapped doesn't need a final T clause
      NIL)
(T (LIST (CONS T (CDR C1]))

```

(SHOULDNT])

(COND.TO.IF

```

[LAMBDA (CONDCLAUSES)
(CONS 'if (CDR (for X in CONDCLAUSES join (COND
      ((AND (EQ (CAR X)
              T)
            (NEQ X (CAR CONDCLAUSES))))
      (CONS 'else (CDR X)))
      (T (CONS 'elseif (CONS (CAR X)
                              (COND
                                ((CDR X)
                                 (CONS 'then (APPEND (CDR X))

```

)

(PUTPROPS LET EDITGETD EDITGETD-LET)

```

(PUTPROPS CL:IF EDITGETD [LAMBDA (FORM)
      (DESTRUCTURING-BIND (TEST THEN ELSE)
        (CDR FORM)
        `(COND (,TEST ,THEN)
              ,@(AND ELSE `(T ,ELSE]))

```

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: NEGATE NEGATE NEGL NEGLST NEGC (NOLINKFNS . T)
(GLOBALVARS NEGATIONS))

(BLOCK%: MKPROGN MKPROGN MKPROGN1 (NOLINKFNS . T))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CLISPARRAY MACROPROPS)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS LAMBDA\$PLST NORMALCOMMENTSFLG COMMENTFLG FIRSTNAME INITIALS INITIALSLST DEFAULTINITIALS)
)

(MOVD? 'NILL 'PREEDITFN)

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

(ADDTOVAR NLAMA)

(ADDTOVAR NLAML MAKECOM CHARMACRO)

(ADDTOVAR LAMA)
)

(PUTPROPS WEDIT COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1988 1990 1991))

FUNCTION INDEX

CHARMACRO4	EDITGETD-LET8	MAKEFN7	NEGC9	SWAPPEDCOND10
COND.TO.IF11	FIRSTATOM5	MKPROGN9	NEGL9	
EDITGETD7	INTCHECK4	MKPROGN19	NEGLST9	
EDITGETD-LAMBDA . . .8	MAKECOM10	NEGATE8	SETTERMCHARS3	

VARIABLE INDEX

CHANGESARRAY2	EDITEMBEDTOKEN2	EDITSMASHUSERFN2	MAXLOOP2
COMPACTHISTORYCOMS5	EDITLOADFNSFLG2	EDITUNSAVEBLOCKFLG2	NEGATIONS4
DONTSAVEHISTORYCOMS .2,3,5	EDITMACROS2,5,6	EDITUSERFN2	UPFINDFLG2
EDITCHARACTERS4	EDITOPS1	FINDFLAG2	USERMACROS2,6
EDITCOMSA2,3,5,7	EDITQUIETFLG2	HISTORYCOMS2	
EDITCOMSL2,3,7	EDITTRACEFN2	MAXLEVEL2	

PROPERTY INDEX

CL:IF11	LET11
-------------------	-----------------
