

File created: 20-Jan-93 13:48:30 {DSK}<python>lde>lispcore>library>DEDITPP.;2

previous date: 4-Sep-91 13:23:07 {DSK}<python>lde>lispcore>library>DEDITPP.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1986, 1990, 1991, 1993 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ DEDITPPCOMS

```
([COMS
  (DECLARE%: EVAL@COMPILE DONTCOPY
    (FILES (LOADCOMP)
            DSPRINTDEF))
  (INITRECORDS DEDITMAP)
  (FNS DEDIT.LPEND DEDIT.RPSTART MAKEMAPENTRY \DEDITFONT# DSPDSFOR SHOWDEDITMAP)
  (DECLARE%: DONTEVAL@LOAD DOCOPY (P (DEFPRINT 'DEDITMAP 'SHOWDEDITMAP]
  (COMS
    (FNS DEPRINTDEF DEDIT-MAKE-READER-ENV)
    (FNS REPP REPPCHANGES REPPUNRAVEL REPPDELETE REPPINSERT REPPTANGLEDP LEADSPACE SPACINGRULE UNPP
      NXTUSEDX ONELINEP)
    (FNS MOVEDSMAP ADJUSTXTAIL ADJUSTYTAIL ADJDEEXTENT DSLINEFONT DSLINEFONT1 MAXFONT)
    (FNS REFRESHIF REFRESHIF1)
    (FNS COMMENTP HIPT LOWPT WIPE)
    (FNS RESETCLIP)))
  ; DEDITMAP record and accessors
  ; DEDITMAP record definition is on DSPRINTDEF because it is
  ; needed there
  ; DEDIT entry and incremental reprettyprinting
```

:: DEDITMAP record and accessors

```
(DECLARE%: EVAL@COMPILE DONTCOPY
```

```
(FILESLOAD (LOADCOMP)
  DSPRINTDEF)
)
```

```
(/DECLAREDATATYPE 'DEDITMAP
  ' (BYTE POINTER BYTE POINTER WORD WORD WORD WORD FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER)
  ;; ---field descriptor list elided by lister---
  ' 12)
```

```
(DEFINEQ
```

(DEDIT.LPEND

```
[LAMBDA (MAPE)
```

(* bvm%: "22-May-86 12:46")

::: Xpos of start of expression, following the open paren (or wrapper)

```
(IPLUS (fetch STARTX of MAPE)
  (LET ((WRAP (fetch WRAPPER of MAPE))
        (FNT (fetch FNT of MAPE)))
    (COND
      (WRAP (STRINGWIDTH WRAP FNT))
      (T (CHARWIDTH (CHARCODE %)
                    (fetch FNT of MAPE)]
```

(DEDIT.RPSTART

```
[LAMBDA (MAPE)
```

(* bvm%: "22-May-86 12:46")

::: Xpos where expression ends and right paren starts

```
(IDIFFERENCE (fetch STOPX of MAPE)
  (COND
    ((fetch WRAPPER of MAPE)
     0)
    (T (CHARWIDTH (CHARCODE %)
                  (fetch FNT of MAPE)]
```

(MAKEMAPENTRY

```
[LAMBDA (TAIL BACK SX SY EX EY FN)
```

(* hdj "19-Jul-85 11:35")

:: Used to check for existing hashlink and do something fancy. Now should not happen except from dummy blocks.

```
(PUTHASH TAIL
  (create DEDITMAP
    BP _ BACK
    TAIL _ TAIL
    STARTX _ SX
    STARTY _ SY
    STOPX _ EX
    STOPY _ EY
```

```

D# _ (COND
      (BACK (fetch D# of BACK))
      (T (DSPDSFOR)))
F# _ FN)
\DEDITMEHASH])

```

(\DEDITFONT#

```

[LAMBDA NIL
  (OR \DEDITFONT# (SETQ \DEDITFONT# (PROG (FONT FONTTYPE FONT#)
      (SETQ FONT (DSPFONT))
      (SETQ FONTTYPE (fetch (FONTDESCRIPTOR FONTDEVICE) of FONT))
      [OR \DEDITFONTS (SETQ \DEDITFONTS (FONTMAPARRAY NIL 'DISPLAY)
      [SETQ FONT# (for I to (ARRAYSIZE \DEDITFONTS)
          thereis (EQ FONT (COND
              ((EQ FONTTYPE 'DISPLAY)
                (fetch (FONTCLASS DISPLAYFD)
                      of (ELT \DEDITFONTS I)))
              (T (FONTCLASSCOMPONENT (ELT
                                      \DEDITFONTS
                                      I)
                                      FONTTYPE NIL T]
              (RETURN FONT#))))))
    (SHOULDNT])

```

(DSPDSFOR

```

[LAMBDA (DS)
  [OR DS (SETQ DS (GETSTREAM NIL 'OUTPUT)
  (PROG [(V (OR [for I to (ARRAYSIZE \DEDITDSPS) thereis (OR (NOT (STREAMP (ELT \DEDITDSPS I)))
      (EQ DS (ELT \DEDITDSPS I]
    (bind [NU _ (ARRAY (ITIMES 2 (ARRAYSIZE \DEDITDSPS) for J to (ARRAYSIZE \DEDITDSPS)
      do (SETA NU J (ELT \DEDITDSPS J)) finally (SETQ \DEDITDSPS NU)
      (RETURN J]
    (SETA \DEDITDSPS V DS)
    (RETURN V])

```

(SHOWDEDITMAP

```

[LAMBDA (ME)
  (CONS [APPLY 'CONCAT (APPEND (LIST "{")
    [bind V TL (Q _ (CAR (fetch TAIL of ME))) while (LISTP Q)
      do (push V "(")
        (push TL (COND
          ((CDR Q)
            ' " --")
          (T "")))
          (SETQ Q (CAR Q))
        finally (RETURN (COND
          (V (APPEND V (CONS Q TL)))
          (T (LIST Q]
    (LIST " @ " (CONCAT "<" (fetch STARTX of ME)
      " "
      (fetch STARTY of ME)
      " _ "
      (fetch STOPX of ME)
      " "
      (fetch STOPY of ME)
      ">"))
    (LIST (COND
      ((UNPURGEDP ME)
        "}")
      (T " PURGED}"])
    (PACK])

```

(DECLARE%: DONTEVAL@LOAD DOCOPY

(DEFPRINT 'DEDITMAP 'SHOWDEDITMAP)

:: DEDIT entry and incremental reprettyprinting

(DEFINEQ

(DEPRINTDEF

[LAMBDA (TAIL LEFT FONT FILE) ; Edited 4-Sep-91 11:56 by jds

;;; The central pretty-printer for DEDIT -- prints TAIL to FILE with indicated LEFT margin and FONT. TAIL is either an expression, or a map entry whose
;;; TAIL we should start printing with.

:: JDS 8/27/91: Bind FNSLST and FORMFLG, because they're used freely in the pretty printer.

```

(SETQ FILE (GETSTREAM FILE 'OUTPUT))
(WITH-READER-ENVIRONMENT (OR (WINDOWPROP FILE 'READER-ENVIRONMENT)
  (DEDIT-MAKE-READER-ENV)))

```

[LET ((MAKEMAP T)

```

(%#RPARS NIL)
(FNSLST NIL)
(FORMFLG NIL)
(FIRSTPOS (DSPXPOSITION NIL FILE))
(LEFT (DSPXPOSITION NIL FILE))
(RMARGIN (DSPRIGHTMARGIN NIL FILE))
(FILEFLG NIL)
(COMMENTCOL NIL)
(DECLARE (SPECVARS MAKEMAP %#RPARS FNSLST FORMFLG FIRSTPOS LEFT RMARGIN FILEFLG COMMENTCOL))
(RESETLST
  (RESETSAVE **COMMENT**FLG NIL)
  (RESETSAVE (OUTPUT FILE))
  (SETQ \DEDITFONTS (FONTMAPARRAY NIL 'DISPLAY))
  [COND
    ((type? DEDITMAP TAIL)
     (SETQ MAKEMAP (OR (fetch BP of TAIL)
                       T))
      (OR FILE (OUTPUT (fetch PDSP of TAIL)))
      [OR FONT (SETQ FONT (fetch FNT of (COND
                                     ((NEQ MAKEMAP T)
                                      MAKEMAP)
                                     (T TAIL)
                                     (OR LEFT (SETQ LEFT (fetch STARTX of TAIL)))
                                     (SETQ TAIL (fetch TAIL of TAIL)
                                     (PROG ((FIRSTPOS (DSPLEFTMARGIN))
                                             [RMARGIN (IPLUS (DSPLEFTMARGIN)
                                                             (IDIFFERENCE (fetch (REGION WIDTH) of (WINDOWPROP FILE 'REGION))
                                                             (ITIMES 2 (WINDOWPROP FILE 'BORDER))
                                                             COMMENTCOL FNSLST TAILFLG FILEFLG CHANGEFLG (FORMFLG T))
                                             (SETFONT FONT FILE)
                                             (DSPXPOSITION LEFT FILE)
                                             (SUPERPRINT (CAR TAIL)
                                                           TAIL NIL FILE)))]))
      (GETME4 TAIL T])

```

(DEDIT-MAKE-READER-ENV

[LAMBDA (EXPR) ; Edited 26-Nov-86 16:38 by bvm:
;; Creates a READER-ENVIRONMENT object to control the editing environment of EXPR. For now, just use the current environment

```

(MAKE-READER-ENVIRONMENT (AND ATM (CL:SYMBOLP ATM)
                              (NEQ ATM 'NOBIND)
                              (CL:SYMBOL-PACKAGE ATM])
)

```

(DEFINEQ

(REPP

```

[LAMBDA (ENT) (* bvm%: "9-Jun-86 17:11")
  (bind DS OLDE do (SETQ OLDE ENT) ; Save current value
    (SETQ DS (fetch PDSP of ENT))
    [COND
      [(fetch BP of ENT)
       ;; Subexpression -- move to where it starts now, then reprint the expression with clipping region set to confine the
       ;; printing to the space now available
       (MOVETO (fetch STARTX of ENT)
               (fetch STARTY of ENT)
               DS)
       (RESETFORM (RESETCLIP (CONS DS (UNPP ENT))))
       (SETQ ENT (DEPRINTDEF ENT NIL NIL DS)
               ; Reprint the entire window
               (RETURN (SETDEDITMAP DS (fetch TAIL of ENT)
                               (MOVEDSMAP ENT (fetch STOPX of OLDE)
                                       (fetch STOPY of OLDE)
                                       (fetch STOPX of ENT)
                                       (fetch STOPY of ENT)]))

```

(REPPCHANGES

```

[LAMBDA (UL) (* bas%: "12-Sep-84 13:57")
  (for I
    in (bind CL TEM for UE in (SETQ UL (REPPUNRAVEL UL))
      when [AND (SETQ TEM (GETME4 (CAR UE)))
                (PROG (SCR (BK (OR (fetch BP of TEM)
                                   TEM)
                               (OLDCAR (CADR UE))
                               (OLDCCR (CDDR UE))
                               (NEWCAR (CAAR UE))
                               (NEWCCR (CDAR UE))))
              (RETURN (COND
                        ((NEQ (NLISTP NEWCAR)
                               (NLISTP OLDCAR))
                         (SETQ TEM BK))
                        ((EQ NEWCCR OLDCDR)

```

```

                (NEQ OLD CAR NEW CAR))
            ((REPPTANGLEDP (CAR UE)
                UL)
            (SETQ TEM BK))
            ((for I in CL thereis (DOMINATE? I TEM))
            (SETQ TEM BK))
            (T (OR (SELECTQ (SETQ SCR (REPPINSERT TEM OLD CDR NEW CDR))
                (NIL NIL)
                (T (SELECTQ (SETQ SCR (REPPDELETE TEM OLD CDR NEW CDR))
                    (NIL NIL)
                    (T (SETQ TEM BK))
                    (SETQ TEM SCR)))
                (SETQ TEM SCR))
            (NEQ OLD CAR NEW CAR]
        unless (for I in CL thereis (DOMINATE? I TEM)) do (push CL TEM) finally (RETURN CL))
    when (UNPURGEDP I) do ;; Earlier elements of CL may dominate later ones. If so, the latter will be purged by the former's REPP.
        (REPP I])

```

(REPPUNRAVEL

(* bas%: "25-JUL-82 21:05")

```

[LAMBDA (UL)
;; Reverses and unpacks LISPXHIST entries
(PROG (RSLT)
  LP [COND
    (NULL UL)
    (RETURN RSLT))
  [(EQ 'LISPXHIST (CAAR UL))
  (for I in (CDAR UL) do (COND
    ((LISTP (CAR I))
    (push RSLT I))
    (EQ (CAR I)
    '/RPLACA)
    (push RSLT (CONS (CADR I)
                    (CONS (CADDR I)
                        (CDADR I]
    (T (push RSLT (CAR UL]
    (SETQ UL (CDR UL))
    (GO LP])

```

(REPPDELETE

(* bas%: " 7-Mar-84 18:09")

```

[LAMBDA (ENT OADR NADR)
  (PROG ([EDGE (for I on OADR thereis (EQ NADR (CDR I)
    NCE OCE SX SY)
  (COND
    [(SETQ EDGE (GETME4 EDGE))
    (SETQ OCE (GETME4 OADR I))
    [AND NADR (SETQ NCE (GETME4 NADR (fetch BP of EDGE)))
    (COND
      ((COMMENTP (CAR NADR))
      (SETQ SX (fetch STARTX of NCE]
    [bind IM for I on OADR until (EQ I NADR) when (SETQ IM (GETME4 I))
    do (COND
      ((COMMENTP (CAR I))
      (UNPP IM))
      (T (COND
        ((COMMENTP (CAR NADR))
        (UNPP IM)))
        (OR SX (SETQ SX (fetch STARTX of IM)))
        (OR SY (SETQ SY (fetch STARTY of IM]
    (RETURN (COND
      [NADR [AND SX (DPCDRSEL NCE)
      (add SX (WIDTH DOTSTRING (fetch FNT of NCE]
      (MOVEDSMAP ENT (fetch STARTX of NCE)
      (fetch STARTY of NCE)
      (OR SX (fetch STARTX of NCE))
      (OR SY (IDIFFERENCE (fetch STOPY of ENT)
        (IDIFFERENCE (fetch STOPY of EDGE)
          (fetch STARTY of NCE]
      (T (MOVEDSMAP ENT (fetch STOPX of EDGE)
        (fetch STOPY of EDGE)
        (fetch STOPX of ENT)
        (fetch STOPY of ENT]
    (T (RETURN T])

```

(REPPINSERT

(* bas%: " 7-MAR-83 09:31")

```

[LAMBDA (ENT OADR NADR)
  (COND
    [(AND (LISTP NADR)
    (OR (NULL OADR)
    (TAILP OADR NADR))
    (PROG ((EDS (fetch PDSP of ENT))
      (ALIGN (SPACINGRULE (fetch BP of ENT)))
      (DELTA (CHARWIDTH (CHARCODE SPACE)

```

```

      (fetch FNT of ENT)))
    (SX (fetch STOPX of ENT))
    (SY (fetch STOPY of ENT))
    NX NY TMP) ; Doesnt enter PROG unless its an insertion
  [SETQ ALIGN (COND
    (ALIGN (fetch STARTX of ALIGN))
    (T (IPLUS DELTAX SX]
  (RESETFORM (RESETCLIP (CONS EDS
    (create REGION
      LEFT _ SX
      BOTTOM _ SY
      WIDTH _ 0
      HEIGHT _ 0)))
    (MOVETO SX SY EDS)
    (for E on NCDR until (EQ E OCDR) first (SETQ TMP ENT)
      do (LEADSPACE E TMP ALIGN DELTAX EDS)
        (SETQ TMP (DEPRINTDEF E (DSPXPOSITION NIL EDS)
          (fetch FNT of (fetch BP of ENT))
          EDS))
        (replace BP of TMP with (fetch BP of ENT))
        finally (LEADSPACE OCDR TMP ALIGN DELTAX EDS)))
    (SETQ NX (DSPXPOSITION NIL EDS))
    (SETQ NY (DSPYPOSITION NIL EDS))
    [PROG (NSY (QV (GETME4 OCDR)))
      (COND
        (QV (SETQ NSY (fetch STARTY of QV))
          (SETQ SX (fetch STARTX of QV))
          (COND
            ((ILESSP NSY SY)
              (REFRESHIF EDS (HIPT ENT)
                (ADD1 (HIPT QV)))
              (SETQ SY NSY))
            ((EQ NSY SY)
              (SETQ NX (IMAX NX SX]
              (RETURN (MOVEDSMAP TMP SX SY NX NY]
    (T T])

```

; Some action at the end of ENT's line?

; Dont move if insert did not reach rest of line eg a comment

(REPPTANGLEDP

```

[LAMBDA (E L) ; (* bas%: " 3-Dec-84 21:45")
  ;; Can only handle one change per cell because of cancelling changes or one CDR change per command lest different CDR changes share
  ;; elements
  (bind EC CCC for I in L when (GETME4 (CAR I)) do (COND
    ((NEQ E (CAR I)))
    (EC (RETURN T))
    (T (SETQ EC T)))
    (COND
      ((EQ (CDAR I)
        (CDDR I)))
      (CCC (RETURN T))
      (T (SETQ CCC T])

```

(LEADSPACE

```

[LAMBDA (E PRV ALIGN DELTAX EDS) ; (* bas%: " 3-DEC-82 18:40")
  (COND
    ((NOT E))
    ([AND (LISTP (CAR (fetch TAIL of PRV)))
      (NOT (COMMENTP (CAR E]
      (MOVETO ALIGN (IPLUS (DSPYPOSITION NIL EDS)
        (DSPLINEFEED NIL EDS))
        EDS))
    (T (RELMOVETO DELTAX 0 EDS])

```

(SPACINGRULE

```

[LAMBDA (BME) ; (* bas%: "12-Sep-84 10:46")
  ;; Looks for someone who might know what the current left margin is and returns that someone.
  (bind P Q for E on (fetch SELEXP of BME) unless (COMMENTP (CAR E)) when (SETQ Q (GETSELMAP E))
    do (COND
      ((NEQ (fetch STARTY of Q)
        (fetch STARTY of BME))
        (RETURN Q))
      (P)
      (T (SETQ P Q)))
    finally (RETURN P])

```

(UNPP

```

[LAMBDA (ENT) ; Edited 11-Jun-90 14:57 by mitani
  ;; Clears region printed by ENT, carefully
  (PROG ((EDS (fetch PDSP of ENT))
    (H (FONTPROP (fetch FNT of ENT)
      'HEIGHT))

```

```

(HI (ADD1 (HIPT ENT)))
(LO (LOWPT ENT))
R)
(SETQ R (DSPCLIPPINGREGION NIL EDS))
(COND
  ((NOT (fetch BP of ENT))
    (WIPE (fetch (REGION LEFT) of R)
          (fetch (REGION BOTTOM) of R)
          (fetch (REGION WIDTH) of R)
          (fetch (REGION HEIGHT) of R)
          EDS)
    (RETURN R))
  ((ONLINEP ENT)
    (WIPE (fetch STARTX of ENT)
          LO
          (IDIFFERENCE (fetch STOPX of ENT)
                       (fetch STARTX of ENT))
          H EDS))
  (T (WIPE (fetch STARTX of ENT)
          (IDIFFERENCE HI H)
          (IDIFFERENCE (fetch (REGION PRIGHT) of R)
                       (fetch STARTX of ENT))
          H EDS)
      ; Amazingly enough this is as good as one can do
      (WIPE (fetch (REGION LEFT) of R)
            (IPLUS LO H)
            (fetch (REGION WIDTH) of R)
            (IDIFFERENCE (IDIFFERENCE HI H)
                         (IPLUS LO H))
            EDS)
      (WIPE (fetch (REGION LEFT) of R)
            LO
            (ADD1 (IDIFFERENCE (fetch STOPX of ENT)
                              (fetch (REGION LEFT) of R)))
            H EDS)))
(RETURN (create REGION
              LEFT _ (fetch (REGION LEFT) of R)
              BOTTOM _ (IMAX LO (fetch (REGION BOTTOM) of R))
              WIDTH _ (COND
                ((ONLINEP ENT)
                  (IDIFFERENCE (NXTUSEDX ENT)
                              (fetch (REGION LEFT) of R)))
                (T (fetch (REGION WIDTH) of R)))
              HEIGHT _ (IMAX 0 (IDIFFERENCE (IMIN HI (fetch (REGION TOP) of R))
                                             (IMAX LO (fetch (REGION BOTTOM) of R))

```

(NXTUSEDX

[LAMBDA (E) ; Edited 11-Jun-90 14:57 by mitani

;; Finds the first used X loc on the same line as the end of E

```

(PROG (V)
  (RETURN (OR [COND
    [(SETQ V (CDR (fetch TAIL of E)))
      (COND
        [(LISTP V)
          (SETQ V (GETME4 V (GETMEBP E)))
          (COND
            ((EQ (fetch STARTY of V)
                 (fetch STOPY of E))
              (fetch STARTX of V]
          (T
            (IPLUS (fetch STOPX of E)
                  (CHARWIDTH (CHARCODE SPACE)
                              (fetch FNT of E]
            ((SETQ V (fetch BP of E))
              (COND
                ((EQ (fetch STOPY of V)
                     (fetch STOPY of E))
                  (fetch RPSTART of V]
            (fetch (REGION RIGHT) of (DSPCLIPPINGREGION NIL (fetch PDSP of E]))

```

; Dotted pair

(ONLINEP

[LAMBDA (ENT) (* bas%: " 4-OCT-82 15:26")

```

(EQ (fetch STARTY of ENT)
    (fetch STOPY of ENT])
)

```

(DEFINEQ

(MOVEDSMAP

[LAMBDA (ENT OX OY NX NY) ; Edited 11-Jun-90 14:57 by mitani

;; APOLOGY: This code and any path by which you got here is a frightful kludge. WARNING: It is also very tricky as there are lots of special cases.

```

(PROG (LOW NLOW FONTH NEXT REG BOTTOM LEFT RIGHT WIDTH (DX (IDIFFERENCE NX OX))
      (DY (IDIFFERENCE NY OY))

```

```

(LINEFONT (DSLIFONT ENT OY))
(EPDS (fetch PDSP of ENT))
(SETQ FONTH (FONTPROP LINEFONT 'DESCENT))
(SETQ NLOW (IDIFFERENCE NY FONTH))
(SETQ OLOW (IDIFFERENCE OY FONTH))
(SETQ FONTH (FONTPROP LINEFONT 'HEIGHT))
(SETQ REG (DSPCLIPPINGREGION NIL EPDS))
(SETQ BOTTOM (fetch (REGION BOTTOM) of REG))
(SETQ LEFT (fetch (REGION LEFT) of REG))
(SETQ RIGHT (fetch (REGION PRIGHT) of REG))
(SETQ WIDTH (fetch (REGION WIDTH) of REG))
[COND
  ((ZEROP DX))
  (T (for (B _ ENT) by (fetch BP of B) while (fetch BP of B)
      do (SETQ NEXT (OR (ADJUSTXTAIL (CDR (fetch TAIL of B))
                          (fetch BP of B)
                          DX OY RIGHT)
                      NEXT))) ; Move the rest of the line

(COND
  ((AND (ILESSP DX 0)
        (IGEQ DY 0))
   (BITBLT EPDS OX OLOW EPDS NX OLOW (IDIFFERENCE RIGHT OX)
           FONTH
           'INPUT
           'REPLACE) ; Move in then blank out far edge
  (WIPE (IPLUS RIGHT DX)
        OLOW
        (IMINUS DX)
        FONTH EPDS))
  (T (WIPE OX OLOW (IDIFFERENCE RIGHT OX)
        FONTH EPDS) ; Image is filled in at exit

[COND
  ((ZEROP DY))
  (T (BITBLT EPDS LEFT BOTTOM EPDS LEFT (IPLUS BOTTOM DY)
      WIDTH
      (IDIFFERENCE OLOW BOTTOM)
      'INPUT
      'REPLACE)
    (for (B _ ENT) by (fetch BP of B) while (fetch BP of B)
      do (ADJUSTYTAIL (CDR (fetch TAIL of B))
                    (fetch BP of B)
                    OY DY)) ; Map over everything to the bottom right moving it vertically
    (ADJDEEXTENT EPDS DY) ; Fix extent and blank inserted space

(COND
  ((IGREATERP DY 0)
   (WIPE LEFT (IMIN BOTTOM OLOW)
           WIDTH DY EPDS) ; Repaint into cleared space
  (REFRESHIF EPDS (IPLUS BOTTOM DY)
             BOTTOM) ; Clear rest of new line
  (WIPE NX NLOW (IDIFFERENCE RIGHT NX)
        FONTH EPDS))
  (T [SETQ NLOW (IMIN NLOW (IPLUS DY (fetch (REGION PTOP) of REG))
    (WIPE LEFT NLOW WIDTH (IMINUS DY)
          EPDS) ; Clear possible trash thru which we extended
    (WIPE LEFT OLOW WIDTH FONTH EPDS]
  (REFRESHIF EPDS (IPLUS FONTH -1 (IMAX NLOW OLOW))
              (ADD1 NLOW))

```

;; Another small kludge. A slightly bigger font like CLISPFONT on the next line might stick up into NLOW and thus get refreshed. Unfortunately, there is no guarantee that that line will be valid to refresh. Correct solution is to make line spacing on printing such that no two lines touch. For now, we diddle the NLOW value to avoid touching the next line down.

```
(RETURN NEXT)]
```

(ADJUSTXTAIL

(* bas%: " 3-Dec-84 22:07")

```

[LAMBDA (TAIL BK DX YLINE RIGHT)
  (PROG (OVER)
    [bind IM for I on TAIL when (SETQ IM (GETME4 I))
      do (COND
          ((NEQ YLINE (fetch STARTY of IM))
           (RETURN))
          ((IGREATERP RIGHT (add (fetch STARTX of IM)
                                DX)))
          (T (SETQ OVER BK)))
        (AND [COND
              ((LISTP (CAR I))
               (ADJUSTXTAIL (CAR I)
                            IM DX YLINE RIGHT))
              ((EQ YLINE (fetch STOPY of IM))
               (ILEQ RIGHT (add (fetch STOPX of IM)
                                DX]
              (SETQ OVER BK))
            finally (COND
              ((SETQ IM (GETME4 I BK))
               (AND (EQ YLINE (fetch STARTY of IM))

```

```

      (ILEQ RIGHT (add (fetch STARTX of IM)
                      DX))
      (SETQ OVER BK))
    (AND (EQ YLINE (fetch STOPY of IM))
          (ILEQ RIGHT (add (fetch STOPX of IM)
                          DX))
          (SETQ OVER BK])
    (AND (EQ YLINE (fetch STOPY of BK))
          (ILEQ RIGHT (add (fetch STOPX of BK)
                          DX))
          (SETQ OVER (OR (fetch BP of BK)
                        BK)))
    (RETURN OVER])

```

(ADJUSTYTAIL

```

[LAMBDA (TAIL BK OY D)
  [bind IM for I on TAIL when (SETQ IM (GETME4 I)) do (add (fetch STARTY of IM)
                                                           D)
  (COND
   ((LISTP (CAR I))
    (ADJUSTYTAIL (CAR I)
                 IM OY D))
   (T (add (fetch STOPY of IM)
           D)))
  finally (COND
           ((SETQ IM (GETME4 I BK))
            (add (fetch STARTY of IM)
                D)
            (add (fetch STOPY of IM)
                D])
           (add (fetch STOPY of BK)
               D])

```

(ADJDEEXTENT

```

[LAMBDA (EX DY)
  (OR (SETQ EX (WINDOWPROP EX 'EXTENT))
      (SHOULDNT))
  (add (fetch (REGION BOTTOM) of EX)
      DY)
  (add (fetch (REGION HEIGHT) of EX)
      (IMINUS DY]))

```

; Edited 11-Jun-90 14:57 by mitani

(DSLINFONT

```

[LAMBDA (E Y)
  (DSLINFONT1 [for old E by (fetch BP of E) thereis (OR (NOT (fetch BP of E))
                                                         (AND (ILESSP Y (fetch STARTY of E))
                                                         (IGREATERP Y (fetch STOPY of E))
                                                         Y))

```

(DSLINFONT1

```

[LAMBDA (ENT YLINE)
  (AND ENT (bind IM (MFONT _ (AND (OR (EQ YLINE (fetch STARTY of ENT))
                                     (EQ YLINE (fetch STOPY of ENT)))
                                 (fetch FNT of ENT))))
  for I on (LISTP (fetch SELEXP of ENT)) do (SETQ MFONT (MAXFONT MFONT (DSLINFONT1 (GETME4 I)
                                                                                   YLINE)))
  finally (RETURN MFONT])

```

(MAXFONT

```

[LAMBDA (F1 F2)
  (COND
   ((IGREATERP (COND
                ((FONTP F1)
                 (FONTPROP F1 'HEIGHT))
                (T 0))
                (COND
                 ((FONTP F2)
                  (FONTPROP F2 'HEIGHT))
                 (T 0)))
    F1)
   (T F2]))

```

)

(DEFINEQ

(REFRESHIF

```

[LAMBDA (WDS HI LO)

```

; Edited 11-Jun-90 14:57 by mitani

;;; Repaints stuff LOWER than HI and on or above LO

```

(WITH-READER-ENVIRONMENT (OR (WINDOWPROP WDS 'READER-ENVIRONMENT)

```



```

      (SHOULDNT))
      (DSPRIGHTMARGIN (PROG1 (DSPRIGHTMARGIN 10000 WDS) ; We reset margin b/c REFRESHIF is sometimes called with
      ; things that would overflow
      [LET ((R (DSPCLIPPINGREGION NIL WDS)))
      (REFRESHIF1 (GETMAP? WDS)
      (GETSTREAM WDS 'OUTPUT)
      (IMIN HI (fetch (REGION PTOP) of R))
      (IMAX LO (fetch (REGION BOTTOM) of R])
      WDS))])

```

(REFRESHIF1

[LAMBDA (M DS HI LO) (* bvm%: "28-May-86 15:30")

;;; Refresh display of that part of expression indicated by map entry M that lies between ypos HI and LO

```

(COND
  ((AND M (OVERLAP HI LO (HIPT M)
    (LOWPT M)))
  (COND
    [(LISTP (fetch SELEXP of M))
    (LET ((WRAP (fetch WRAPPER of M)))
      (COND
        ([IGREATERP HI (IDIFFERENCE (fetch STARTY of M)
          (FONTPROP (fetch FNT of M)
            'DESCENT])
          (MOVETO (fetch STARTX of M)
            (fetch STARTY of M)
            DS)
          (DSPFONT (fetch FNT of M)
            DS)
          (PRIN3 (OR WRAP '%)
            DS))
        (COND
          (WRAP (REFRESHIF1 (GETME4 (CDR (fetch SELEXP of M))
            M)
            DS HI LO))
          (T [for I on (fetch SELEXP of M) do (REFRESHIF1 (GETME4 I M)
            DS HI LO)
            finally (COND
              (I (SETQ I (GETME4 I M))
                (MOVETO (IDIFFERENCE (fetch STARTX of I)
                  (STRINGWIDTH DOTSTRING (fetch FNT of M)))
                  (fetch STARTY of I)
                  DS)
                (PRIN3 DOTSTRING DS) ; Dotted pair
                (REFRESHIF1 I DS HI LO]
            (COND
              ([ILEQ LO (IPLUS (fetch STOPY of M)
                (FONTPROP (fetch FNT of M)
                  'ASCENT])
                (MOVETO (fetch RPSTART of M)
                  (fetch STOPY of M)
                  DS)
                (DSPFONT (fetch FNT of M)
                  DS)
                (PRIN3 '%) DS]
              (T (MOVETO (fetch STARTX of M)
                (fetch STARTY of M)
                DS)
                (DSPFONT (fetch FNT of M)
                  DS)
                (COND
                  ((fetch LONGSTRINGP of M)
                  (LET* ((COMMENTP (NULL (fetch LONGSTRINGLMARGIN of M)))
                    [LMARG (fetch STARTX of (COND
                      (COMMENTP ; Inside a comment, the string may be printed with a margin to
                      ; the left of where the string starts
                      (fetch BP of M))
                    (T M]
                    (RMARG (WINDOWPROP DS 'WIDTH))
                    MAKEMAP)
                    (DECLARE (SPECVARS MAKEMAP))
                    (PRIN2-LONG-STRING (fetch SELEXP of M)
                      DS T T LMARG (COND
                        ((fetch LONGSTRINGSYMMETRICP of M)
                          ; String symmetrically centered
                          (IDIFFERENCE RMARG LMARG))
                        (T RMARG))
                    COMMENTP)))
                  (T (PRIN4 (fetch SELEXP of M)
                    DS])
                )
            )
          )
    )
  (DEFINEQ

```

(COMMENTP

[LAMBDA (E)
(AND COMMENTFLG (EQ COMMENTFLG (CAR (LISTP E))

(* bas%: "15-NOV-82 22:01")

(HIPT

[LAMBDA (ENT)
(IPLUS (fetch STARTY of ENT)
(FONTPROP (fetch FNT of ENT)
'ASCENT)
-1])

(* bas%: " 4-OCT-82 15:25")

(LOWPT

[LAMBDA (E)
(IDIFFERENCE (fetch STOPY of E)
(FONTPROP (fetch FNT of E)
'DESCENT])

(* bas%: " 4-OCT-82 15:25")

(WIPE

[LAMBDA (X Y W H DS)
(BITBLT NIL NIL NIL DS X Y W H 'TEXTURE 'REPLACE (DSPTEXTURE NIL DS])

(* bas%: "19-AUG-82 15:18")

)

(DEFINEQ

(RESETCLIP

[LAMBDA (C)
;; For use in RESETFORM. Takes a CONS of a DSP and its new region
(CONS (CAR C)
(DSPCLIPPINGREGION (CDR C)
(CAR C])

(* bas%: " 8-NOV-82 15:35")

)

(PUTPROPS DEDITPP COPYRIGHT ("Venue & Xerox Corporation" 1986 1990 1991 1993))

FUNCTION INDEX

ADJDEEXTENT8	DSLINFONT18	ONELINEP6	RESETCLIP10
ADJUSTXTAIL7	DSPDSFOR2	REFRESHIF8	SHOWDEDITMAP2
ADJUSTYTAIL8	HIPT10	REFRESHIF19	SPACINGRULE5
COMMENTP10	LEADSPACE5	REPP3	UNPP5
DEDIT-MAKE-READER-ENV	...3	LOWPT10	REPPCHANGES3	WIPE10
DEDIT.LPEND1	MAKEMAPENTRY1	REPPDELETE4	\DEDITFONT#2
DEDIT.RPSTART1	MAXFONT8	REPPINSERT4		
DEPRINTDEF2	MOVEDSMAP6	REPPTANGLEDP5		
DSLINFONT8	NXTUSEDX6	REPPUNRAVEL4		
