File created: 26-Dec-2021 18:59:24 **{DSK}<Users>kaplan>Local>medley3.5>my-medley>lispusers>OBJECTWINDOW.;5**

changes to: (FNS OBJ.CREATEW)

previous date: 21-Dec-2021 18:20:31 {DSK}<Users>kaplan>Local>medley3.5>my-medley>lispusers>OBJECTWINDOW.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ OBJECTWINDOWCOMS
       [(DECLARE%: DOEVAL@LOAD DONTCOPY (RECORDS OBJ))

;;; User callable functions

        (FNS OBJ.ADDMANYTOW OBJ.ADDTOW OBJ.CLEARW OBJ.CREATEW OBJ.DELFROMW OBJ.FIND.REGION OBJ.INSERTOBJECTS
             OBJ.MAP.OBJECTS OBJ.OBJECTS OBJ.REPLACE OBJWINDOWP)

;;; Routines called by user routines

        (FNS OBJ.APPLY.USER.FN OBJ.BUTTONEVENTFN OBJ.BUTTONEVENTINFN OBJ.CLEAR.EXTENT OBJ.COMPUTE.IMAGEBOX
             OBJ.COMPUTE.REGION OBJ.COPYBUTTONEVENTFN OBJ.DELFROMW.HORIZONTAL OBJ.DELFROMW.VERTICAL
             OBJ.DRAW.OBJECT OBJ.END.OF.OBJECT OBJ.FIND.OBJECT OBJ.FIND.REGION.HORIZONTAL
             OBJ.FIND.REGION.VERTICAL OBJ.FLIP.OBJECT OBJ.HARDCOPYFN OBJ.INDEX.OBJECT OBJ.INSTANTIATE
             OBJ.MOVETO.LAST.INSTANTIATED.OBJECT OBJ.RECOMPUTE.EXTENT OBJ.REPAINTFN OBJ.REPLACE.HORIZONTAL
             OBJ.REPLACE.VERTICAL OBJ.RESHAPEFN OBJ.SCROLLFN OBJ.SCROLLFN.HORIZONTAL OBJ.SCROLLFN.VERTICAL)
        (P (AND (GETD 'MODERNWINDOW.SETUP)
                (MODERNWINDOW.SETUP (FUNCTION OBJ.BUTTONEVENTINFN])

(DECLARE%: DOEVAL@LOAD DONTCOPY

(DECLARE%: EVAL@COMPILE

[RECORD OBJ (OBJECT REGION YDESC XKERN INSTANTIATED)
       (ACCESSFNS ((ASCENT (IDIFFERENCE (fetch (REGION HEIGHT) of (fetch (OBJ REGION) of DATUM))
                                        (fetch (OBJ YDESC) of DATUM]
)
)

;;; User callable functions

(DEFINEQ

(OBJ.ADDMANYTOW
  [LAMBDA (WINDOW OBJECTS)                                    ; Edited 21-Dec-2021 18:20 by rmk
                                                              ; Edited  3-Aug-93 09:30 by rmk:
                                                              (* bbb " 7-Jan-86 16:15")

;;; For the moment this is just like calling OBJ.ADDTOW for each object in OBJECTS

    (FOR OBJECT INSIDE OBJECTS DO (OBJ.ADDTOW WINDOW OBJECT])

(OBJ.ADDTOW
  [LAMBDA (WINDOW OBJECT)                                     ; Edited  3-Aug-93 09:30 by rmk:
                                                              (* bbb "19-Dec-85 11:37")

;;; OBJECT is added to the property value OBJECTS of WINDOW at the current position in WINDOW The objects in OBJECT are ordered by their
;;; leading edge.  The window is redrawn if necessary.

    (LET* ((WINDOWTYPE (WINDOWPROP WINDOW 'WINDOWTYPE))
           (OBJECTS (WINDOWPROP WINDOW 'OBJECTS))
           (ADDED.OBJECT (CREATE OBJ
                                 OBJECT _ OBJECT
                                 INSTANTIATED _ NIL))
           [POINT.MOVED (OR (NEQ (DSPXPOSITION NIL WINDOW)
                                 (WINDOWPROP WINDOW 'OLDXPOSITION))
                            (NEQ (DSPYPOSITION NIL WINDOW)
                                 (WINDOWPROP WINDOW 'OLDYPOSITION]
           [POINT.BEFORE.END.OF.CLIPPING.REGION (IF (EQ WINDOWTYPE 'HORIZONTAL)
                                                    THEN (ILESSP (DSPXPOSITION NIL WINDOW)
                                                                 (FETCH (REGION RIGHT) OF (DSPCLIPPINGREGION
                                                                                           NIL WINDOW)))
                                                  ELSE (IGREATERP (DSPYPOSITION NIL WINDOW)
                                                                  (FETCH (REGION BOTTOM) OF (DSPCLIPPINGREGION
                                                                                             NIL WINDOW]
           (LASTOBJECTS))
          (COND
             ((AND (NULL OBJECTS)
                   (NOT POINT.MOVED))
                ;; When the window was created the x and y positions were unspecified.  Now we will resolve them if the user hasn't for us.
```

```
            (OBJ.COMPUTE.IMAGEBOX WINDOW ADDED.OBJECT)
            (DSPXPOSITION 0 WINDOW)
            (DSPYPOSITION (IDIFFERENCE (FETCH (REGION TOP) OF (DSPCLIPPINGREGION NIL WINDOW))
                                       (FETCH (OBJ ASCENT) OF ADDED.OBJECT))
                   WINDOW)
            (OBJ.COMPUTE.REGION WINDOW ADDED.OBJECT))
           ((OR POINT.BEFORE.END.OF.CLIPPING.REGION POINT.MOVED)
            (OBJ.COMPUTE.IMAGEBOX WINDOW ADDED.OBJECT)
            (IF (EQ WINDOWTYPE 'VERTICAL)
                THEN (RELMOVETO 0 (IMINUS (FETCH (OBJ ASCENT) OF ADDED.OBJECT))
                                  WINDOW))
            (OBJ.COMPUTE.REGION WINDOW ADDED.OBJECT)))        ; Insert the new object in the list which is in order of leading edge
      (IF (NULL OBJECTS)
          THEN (WINDOWPROP WINDOW 'OBJECTS (LIST ADDED.OBJECT))
        ELSEIF [OR [AND (EQ WINDOWTYPE 'HORIZONTAL)
                        (ILESSP (DSPXPOSITION NIL WINDOW)
                                (FETCH (REGION LEFT) OF (FETCH (OBJ REGION) OF (CAR OBJECTS]
                   (AND (EQ WINDOWTYPE 'VERTICAL)
                        (IGREATERP (DSPYPOSITION NIL WINDOW)
                                   (FETCH (REGION TOP) OF (FETCH (OBJ REGION) OF (CAR OBJECTS]
          THEN (ATTACH ADDED.OBJECT OBJECTS)
        ELSEIF POINT.MOVED
          THEN (BIND SECOND.OBJECT FOR OBJECTTAIL ON OBJECTS
                  DO (SETQ SECOND.OBJECT (CADR OBJECTTAIL))
                     (IF SECOND.OBJECT
                         THEN (OBJ.INSTANTIATE WINDOW SECOND.OBJECT (CAR OBJECTTAIL))
                              (IF (EQ WINDOWTYPE 'HORIZONTAL)
                                  THEN (IF (IGREATERP (DSPXPOSITION NIL WINDOW)
                                                      (FETCH (REGION LEFT) OF (FETCH (OBJ REGION)
                                                                                     OF ADDED.OBJECT)))
                                           THEN (RPLACD OBJECTTAIL (CONS ADDED.OBJECT (CDR OBJECTTAIL)))
                                                (RETURN))
                                ELSE (IF (ILESSP (DSPYPOSITION NIL WINDOW)
                                                 (FETCH (REGION BOTTOM) OF (FETCH (OBJ REGION)
                                                                                  OF ADDED.OBJECT)))
                                         THEN (RPLACD OBJECTTAIL (CONS ADDED.OBJECT (CDR OBJECTTAIL)))
                                              (RETURN)))
                           ELSE (RPLACD OBJECTTAIL (LIST ADDED.OBJECT))
                                (RETURN))                     ; At the end

                  )
        ELSE (SETQ LASTOBJECTS (LAST OBJECTS))
             (IF POINT.BEFORE.END.OF.CLIPPING.REGION
                 THEN (OBJ.INSTANTIATE WINDOW ADDED.OBJECT (CAR LASTOBJECTS)))
             (RPLACD LASTOBJECTS (LIST ADDED.OBJECT)))

      ;; Remember the old x and y, draw the object then reposition the x or y to be ready for adding the next object.
      (OBJ.RECOMPUTE.EXTENT WINDOW)
      (IF (AND (FETCH (OBJ INSTANTIATED) OF ADDED.OBJECT)
               (REGIONSINTERSECTP (DSPCLIPPINGREGION NIL WINDOW)
                      (FETCH (OBJ REGION) OF ADDED.OBJECT)))
          THEN (OBJ.DRAW.OBJECT WINDOW ADDED.OBJECT))
      (OBJ.MOVETO.LAST.INSTANTIATED.OBJECT WINDOW (WINDOWPROP WINDOW 'OBJECTS))
                                                              ; Finally move the point to after the last instantiated object.
      (WINDOWPROP WINDOW 'OLDXPOSITION (DSPXPOSITION NIL WINDOW))
      (WINDOWPROP WINDOW 'OLDYPOSITION (DSPYPOSITION NIL WINDOW))
      OBJECT])
```

(OBJ.CLEARW

```
  [LAMBDA (WINDOW)                                            (* rmk%: "17-Feb-88 10:19")
                                                              (* bbb "13-May-86 15:15")
    (if (WINDOWPROP WINDOW 'OBJECTS NIL)
        then

          (* Don't clear it if there aren't any objects. Stops a NOOPEN window from popping up when it's created.)

          (CLEARW WINDOW))
    (if (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
            'VERTICAL)
        then (WYOFFSET (SUB1 (WINDOWPROP WINDOW 'HEIGHT))
                       WINDOW)                                (* In vertical windows the top of the window has Y = 0)
          )
    (OBJ.CLEAR.EXTENT WINDOW)
    (DSPXPOSITION MIN.FIXP WINDOW)

          (* Changed the x and y position to min and max FIXP from min and max INTEGER)

    (DSPYPOSITION MAX.FIXP WINDOW)
    (DSPRIGHTMARGIN 65535 WINDOW)
    (WINDOWPROP WINDOW 'OLDXPOSITION (DSPXPOSITION NIL WINDOW))
    (WINDOWPROP WINDOW 'OLDYPOSITION (DSPYPOSITION NIL WINDOW))
    WINDOW])
```

(OBJ.CREATEW

```
  [LAMBDA (WINDOWTYPE REGION/WINDOW TITLE BORDERSIZE NOOPENFLG SEPDIST BOXFN DISPLAYFN BUTTONINFN HARDCOPYFN
```

```
                HCPYHEADING)                                          ; Edited 26-Dec-2021 18:48 by rmk
                                                                      ; Edited 21-Dec-2021 17:19 by rmk
                                                                      ; Edited 16-Dec-2021 23:32 by rmk
                                                                      ; Edited 26-Nov-96 14:31 by rmk:
                                                                      (* bbb " 9-May-86 16:59")
    (CL:UNLESS (MEMB WINDOWTYPE '(HORIZONTAL VERTICAL))
           (\ILLEGAL.ARG WINDOWTYPE))
    (LET (WINDOW)
         (IF (WINDOWP REGION/WINDOW)
             THEN (SETQ WINDOW REGION/WINDOW)
                  (CL:WHEN TITLE
                      (WINDOWPROP WINDOW 'TITLE TITLE))
           ELSE (SETQ WINDOW (CREATEW REGION/WINDOW TITLE BORDERSIZE NOOPENFLG)))
         (WINDOWPROP WINDOW 'WINDOWTYPE WINDOWTYPE)
         (OBJ.CLEARW WINDOW)
         (WINDOWPROP WINDOW 'SCROLLFN (FUNCTION OBJ.SCROLLFN))
         (WINDOWPROP WINDOW 'REPAINTFN (FUNCTION OBJ.REPAINTFN))
         (WINDOWPROP WINDOW 'RESHAPEFN (FUNCTION OBJ.RESHAPEFN))
         (WINDOWPROP WINDOW 'COPYBUTTONEVENTFN (FUNCTION OBJ.COPYBUTTONEVENTFN))
         (WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION OBJ.BUTTONEVENTFN))
         (WINDOWPROP WINDOW 'SEPARATIONDISTANCE (OR SEPDIST 0))
         (WINDOWPROP WINDOW 'BOXFN BOXFN)
         (WINDOWPROP WINDOW 'DISPLAYFN DISPLAYFN)
         (WINDOWPROP WINDOW 'BUTTONINFN BUTTONINFN)
         [WINDOWPROP WINDOW 'HARDCOPYFN (LIST (OR HARDCOPYFN (FUNCTION OBJ.HARDCOPYFN))
                                              (OR HCPYHEADING 'TITLE]
                                                                      ; Limit the scrolling to the extent depending on the window type
         [WINDOWPROP WINDOW 'SCROLLEXTENTUSE (if (EQ WINDOWTYPE 'HORIZONTAL)
                                                 then '(LIMIT . T)
                                               else '(T . LIMIT]
         WINDOW])
```

(OBJ.DELFROMW

```
  [LAMBDA (WINDOW OBJECT)                                             ; Edited  3-Aug-93 09:28 by rmk:
                                                                      (* bbb "19-Dec-85 17:13")
    (IF (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
            'HORIZONTAL)
        THEN (OBJ.DELFROMW.HORIZONTAL WINDOW OBJECT)
      ELSE (OBJ.DELFROMW.VERTICAL WINDOW OBJECT])
```

(OBJ.FIND.REGION

```
  [LAMBDA (WINDOW SEARCHOBJECT)                                       (* bbb "11-Dec-85 10:01")

          (* The object SEARCHOBJECT is searched for and its region is returned.
          This may involve instantiating objects.)

    (IF (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
            'HORIZONTAL)
        THEN (OBJ.FIND.REGION.HORIZONTAL WINDOW SEARCHOBJECT)
      ELSE (OBJ.FIND.REGION.VERTICAL WINDOW SEARCHOBJECT])
```

(OBJ.INSERTOBJECTS

```
  [LAMBDA (WINDOW NEWOBJECTS OLDOBJECT WHERE)                         ; Edited 21-Dec-2021 18:19 by rmk
                                                                      ; Edited 12-Aug-93 23:01 by rmk:
                                                                      (* bbb "19-Dec-85 11:37")
    ;; NEWOBJECTS are inserted in WINDOW at position WHERE (BEFORE or AFTER) with respect to OLDOBJECT.

    (SETQ NEWOBJECTS (MKLIST NEWOBJECTS))
    (LET* [(WINDOWTYPE (WINDOWPROP WINDOW 'WINDOWTYPE))
           (OBJECTS (WINDOWPROP WINDOW 'OBJECTS))
           (PREVTAIL)
           (OLDOBJTAIL (AND OLDOBJECT (IF (IMAGEOBJP OLDOBJECT)
                                          THEN (FOR OTAIL ON OBJECTS
                                                  DO (IF (EQ OLDOBJECT (FETCH (OBJ OBJECT) OF (CAR OTAIL)))
                                                         THEN (RETURN OTAIL)
                                                       ELSE (SETQ PREVTAIL OTAIL)))
                                        ELSE (MEMB OLDOBJECT OBJECTS]
          (IF (AND OLDOBJTAIL WHERE)
              THEN (SELECTQ WHERE
                        (BEFORE (CL:UNLESS PREVTAIL                   ; If this is the earliest item, insert it at the beginning of the the
                                                                      ; clipping region. Vertical case needs to be thought out.
                                    (DSPXPOSITION (FETCH (REGION LEFT) OF (DSPCLIPPINGREGION NIL WINDOW))
                                           WINDOW))
                                (FOR O IN OLDOBJTAIL DO (REPLACE INSTANTIATED OF O WITH NIL))
                                (FOR O IN NEWOBJECTS DO (ATTACH (CREATE OBJ
                                                                        OBJECT _ O)
                                                                OLDOBJTAIL))
                                (FOR F (PREV _ (CAR PREVTAIL)) IN (OR (CDR PREVTAIL)
                                                                      OBJECTS)
                                   DO (OBJ.INSTANTIATE WINDOW F PREV)
                                      (SETQ PREV F)))
                        (AFTER (FOR O IN (CDR OLDOBJTAIL) DO (REPLACE INSTANTIATED OF O WITH NIL))
                               (FOR O (FOLLOWINGOBJECTS _ (CDR OLDOBJTAIL))
                                    (PREV _ (CAR OLDOBJTAIL))
```

```
                                    (OTAIL _ OLDOBJTAIL) IN NEWOBJECTS DO (SETQ O (CREATE OBJ
                                                                                       OBJECT _ O))
                                                                          (SETQ OTAIL (PUSH (CDR OTAIL)
                                                                                            O))
                                                                          (OBJ.INSTANTIATE WINDOW O PREV)
                                                                          (SETQ PREV O)
                                  FINALLY
                                        ;; Check logic in OBJ.DELFROMW.  Maybe we don't have to instantiate beyond the visible region
                                        (FOR F IN FOLLOWINGOBJECTS DO (OBJ.INSTANTIATE WINDOW F PREV)
                                                                      (SETQ PREV F))))
                             (REPLACE                                 ; Left is left of object being replaced.  Might need to do
                                                                      ; something different for vertical case.
                                      (DSPXPOSITION (FETCH (REGION LEFT) OF (FETCH (OBJ REGION) OF (CAR OLDOBJTAIL)
                                                                                                   ))
                                            WINDOW)
                                      (FOR O IN (CDR OLDOBJTAIL) DO (REPLACE INSTANTIATED OF O WITH NIL))
                                      (RPLACA OLDOBJTAIL (CREATE OBJ
                                                                 OBJECT _ (CAR NEWOBJECTS)))
                                      (OBJ.INSTANTIATE WINDOW (CAR OLDOBJTAIL))
                                      (FOR O (FOLLOWINGOBJECTS _ (CDR OLDOBJTAIL))
                                           (PREV _ (CAR OLDOBJTAIL))
                                           (OTAIL _ OLDOBJTAIL) IN (CDR NEWOBJECTS)
                                         DO (SETQ O (CREATE OBJ
                                                            OBJECT _ O))
                                            (SETQ OTAIL (PUSH (CDR OTAIL)
                                                              O))
                                            (OBJ.INSTANTIATE WINDOW O PREV)
                                            (SETQ PREV O)
                                         FINALLY
                                            ;; Check logic in OBJ.DELFROMW.  Maybe we don't have to instantiate beyond the visible
                                            ;; region
                                            (FOR F IN FOLLOWINGOBJECTS DO (OBJ.INSTANTIATE WINDOW F PREV)
                                                                          (SETQ PREV F))))
                                      (SHOULDNT))
                                (OBJ.RECOMPUTE.EXTENT WINDOW)
                                (REDISPLAYW WINDOW (DSPCLIPPINGREGION NIL WINDOW))
                          ELSE (OBJ.ADDMANYTOW WINDOW NEWOBJECTS))
                    NEWOBJECTS])
```

(OBJ.MAP.OBJECTS

```
  [LAMBDA (WINDOW MAPFN)                                  (* bbb "19-Dec-85 14:39")
```

(* MAPFN is called with the object field of each OBJ in WINDOW If the MAPFN returns non-NIL then this value replaces the object)

```
    (for OBJECT in (WINDOWPROP WINDOW 'OBJECTS) bind FN.RESULT
       do (SETQ FN.RESULT (APPLY* MAPFN (fetch (OBJ OBJECT) of OBJECT)))
          (if FN.RESULT
              then (OBJ.REPLACE WINDOW (fetch (OBJ OBJECT) of OBJECT)
                                FN.RESULT T)))
    (REDISPLAYW WINDOW (DSPCLIPPINGREGION NIL WINDOW)
           T])
```

(OBJ.OBJECTS

```
  [LAMBDA (WINDOW)                                        (* bbb "11-Dec-85 10:42")
```

(* * The list of objects is returned)

```
    (for OBJECT in (WINDOWPROP WINDOW 'OBJECTS) collect (fetch (OBJ OBJECT) of OBJECT])
```

(OBJ.REPLACE

```
  [LAMBDA (WINDOW OLD.OBJECT NEW.OBJECT DONT.REDISPLAY.FLG)       ; Edited 27-Jul-93 17:11 by rmk:
                                                                  (* bbb "19-Dec-85 14:56")
```

;;; Replaces new object with old object and adjusts the region of all objects to its left

```
    (IF (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
            'HORIZONTAL)
        THEN (OBJ.REPLACE.HORIZONTAL WINDOW OLD.OBJECT NEW.OBJECT DONT.REDISPLAY.FLG)
      ELSE (OBJ.REPLACE.VERTICAL WINDOW OLD.OBJECT NEW.OBJECT DONT.REDISPLAY.FLG])
```

(OBJWINDOWP

```
  [LAMBDA (WINDOW)                                        ; Edited  4-May-99 16:27 by rmk:
                                                          ; Edited  4-May-99 16:26 by rmk:
    (AND (WINDOWP WINDOW)
         (EQ 'OBJ.COPYBUTTONEVENTFN (WINDOWPROP WINDOW 'COPYBUTTONEVENTFN))
         (MEMB (WINDOWPROP WINDOW 'WINDOWTYPE)
               '(HORIZONTAL VERTICAL))
         WINDOW])

)
```

;;; Routines called by user routines

```
(DEFINEQ
```

(OBJ.APPLY.USER.FN

```
  [LAMBDA (USER.FN OBJECT WINDOW REG)                    (* jtm%: " 3-Nov-87 17:08")
                                                         ; Edited 28-Jul-93 17:39 by rmk:
```

;;; Sets up the coordinate system and calls the user function (eg. a BUTTONEVENTINFN or a COPYEVENTFN)

```
    (LET* ((WINDOWDISPLAYSTREAM (GETSTREAM WINDOW))
           (RELX (LASTMOUSEX WINDOW))
           (RELY (LASTMOUSEY WINDOW))
           [OBJORIG (OR (IMAGEOBJPROP OBJECT 'OBJECTORIGIN)
                        (CONSTANT (CREATEPOSITION 0 0]
           WINDOWDELTAX WINDOWDELTAY WINDOWCLIPPING.REGION RESULT)
          ;; (IMAGEBOX (APPLY* (IMAGEOBJPROP OBJECT (QUOTE IMAGEBOXFN)) OBJECT WINDOW)) (REG (create REGION LEFT _
          ;; (IDIFFERENCE (DSPXPOSITION NIL DS) (fetch (IMAGEBOX XKERN) of IMAGEBOX)) BOTTOM _ (IDIFFERENCE (DSPYPOSITION
          ;; NIL DS) (fetch (IMAGEBOX YDESC) of IMAGEBOX)) WIDTH _ (fetch (IMAGEBOX XSIZE) of IMAGEBOX) HEIGHT _ (fetch
          ;; (IMAGEBOX YSIZE) of IMAGEBOX)))
          (SETQ WINDOWDELTAX (IDIFFERENCE (OR (IMINUS (fetch (POSITION XCOORD) of OBJORIG))
                                              0)
                                          (fetch (REGION LEFT) of REG)))
          (SETQ WINDOWDELTAY (IDIFFERENCE (OR (IMINUS (fetch (POSITION YCOORD) of OBJORIG))
                                              0)
                                          (fetch (REGION BOTTOM) of REG)))
          (RESETLST
              (RESETSAVE (WXOFFSET (IMINUS WINDOWDELTAX)
                                   WINDOWDISPLAYSTREAM)
                         (LIST (FUNCTION WXOFFSET)
                               WINDOWDELTAX WINDOWDISPLAYSTREAM))
              (RESETSAVE (WYOFFSET (IMINUS WINDOWDELTAY)
                                   WINDOWDISPLAYSTREAM)
                         (LIST (FUNCTION WYOFFSET)
                               WINDOWDELTAY WINDOWDISPLAYSTREAM))
              (SETQ WINDOWCLIPPING.REGION (DSPCLIPPINGREGION NIL WINDOWDISPLAYSTREAM))
              (RESETSAVE (DSPCLIPPINGREGION (INTERSECTREGIONS WINDOWCLIPPING.REGION
                                                              (create REGION
                                                                      LEFT _ (OR (IMINUS (fetch (POSITION XCOORD)
                                                                                                of OBJORIG))
                                                                                 0)
                                                                      BOTTOM _ (OR (IMINUS (fetch (POSITION YCOORD)
                                                                                                  of OBJORIG))
                                                                                   0)
                                                                      WIDTH _ (fetch (REGION WIDTH) of REG)
                                                                      HEIGHT _ (fetch (REGION HEIGHT) of REG)))
                                                   WINDOWDISPLAYSTREAM)
                         (LIST (FUNCTION DSPCLIPPINGREGION)
                               WINDOWCLIPPING.REGION WINDOWDISPLAYSTREAM))
              [ERSETQ (SETQ RESULT (APPLY* USER.FN OBJECT WINDOW '? RELX RELY WINDOW '? '?]
              RESULT)])
```

(OBJ.BUTTONEVENTFN

```
  [LAMBDA (WINDOW STREAM)                                (* bbb "11-Dec-85 10:23")
    (OBJ.BUTTONEVENTINFN WINDOW STREAM])
```

(OBJ.BUTTONEVENTINFN

```
  [LAMBDA (WINDOW STREAM)                                (* jtm%: " 3-Nov-87 17:09")
                                                         ; Edited 28-Jul-93 17:40 by rmk:
```

;;; Determines which object the button was clicked in and calls its BUTTONEVENTINFN. If CHANGED is returned then the region for that object will be redrawn.

```
    (TOTOPW WINDOW)
    (PROG ((CLIPPING.REGION (DSPCLIPPINGREGION NIL WINDOW))
           (MOUSEX (LASTMOUSEX WINDOW))
           (MOUSEY (LASTMOUSEY WINDOW))
           (WINDOWXPOS (DSPXPOSITION NIL WINDOW))
           (WINDOWYPOS (DSPYPOSITION NIL WINDOW))
           RESULT OBJ REG)
      BUTTONDOWN
          [IF (SETQ OBJ (OBJ.FIND.OBJECT WINDOW MOUSEX MOUSEY))
              THEN (SETQ REG (FETCH (OBJ REGION) OF OBJ))
                   (MOVETO (IPLUS (FETCH (OBJ XKERN) OF OBJ)
                                  (FETCH (REGION LEFT) OF REG))
                           (IPLUS (FETCH (OBJ YDESC) OF OBJ)
                                  (FETCH (REGION BOTTOM) OF REG))
                           WINDOW)
                   (SETQ RESULT (OBJ.APPLY.USER.FN (IMAGEOBJPROP (FETCH (OBJ OBJECT) OF OBJ)
                                                                 'BUTTONEVENTINFN)
                                                   (FETCH (OBJ OBJECT) OF OBJ)
                                                   WINDOW REG))
```

```
                    (MOVETO WINDOWXPOS WINDOWYPOS WINDOW)
                    (SELECTQ RESULT
                         (CHANGED (REDISPLAYW WINDOW (FETCH (OBJ REGION) OF OBJ)
                                              T))
                         (ALLCHANGED (REDISPLAYW WINDOW))
                         (IF (EQ (CAR (LISTP RESULT))
                                 '*DOFORM*)
                             THEN ;; Function supplies a form to operate on window, but only after all transformations have been undone.
                                  (EVAL (CADR RESULT]
          (GETMOUSESTATE)
          (IF [AND (LASTMOUSESTATE (OR LEFT MIDDLE))
                   (INSIDEP CLIPPING.REGION (SETQ MOUSEX (LASTMOUSEX WINDOW))
                            (SETQ MOUSEY (LASTMOUSEY WINDOW]
              THEN (GO BUTTONDOWN])
```

(OBJ.CLEAR.EXTENT

```
  [LAMBDA (WINDOW)                                          (* bbb " 9-Dec-85 16:33")
    (WINDOWPROP WINDOW 'EXTENT (create REGION
                                       LEFT _ -1
                                       BOTTOM _ -1
                                       WIDTH _ -1
                                       HEIGHT _ -1])
```

(OBJ.COMPUTE.IMAGEBOX

```
  [LAMBDA (WINDOW OBJECT)                                   ; Edited  3-Aug-93 17:46 by rmk:
                                                            (* bbb "10-Dec-85 11:33")
    (LET* [BOXFN.RESULT (IMAGEBOX (IF (IMAGEOBJP (FETCH (OBJ OBJECT) OF OBJECT))
                                      THEN (APPLY* (IMAGEOBJPROP (FETCH (OBJ OBJECT) OF OBJECT)
                                                                 'IMAGEBOXFN)
                                                   (FETCH (OBJ OBJECT) OF OBJECT)
                                                   WINDOW)
                                    ELSE (SETQ BOXFN.RESULT (APPLY* (WINDOWPROP WINDOW 'BOXFN)
                                                                    (FETCH (OBJ OBJECT) OF OBJECT)
                                                                    WINDOW))
                                    ;; If the result of applying the boxfn for the window with the object returns an image object then
                                    ;; replace the object with this image object and compute this new image object's imagebox
                                         (IF (IMAGEOBJP BOXFN.RESULT)
                                             THEN (REPLACE (OBJ OBJECT) OF OBJECT WITH BOXFN.RESULT)
                                                  (APPLY* (IMAGEOBJPROP (FETCH (OBJ OBJECT) OF OBJECT)
                                                                        'IMAGEBOXFN)
                                                          (FETCH (OBJ OBJECT) OF OBJECT)
                                                          WINDOW)
                                           ELSE BOXFN.RESULT]
          (REPLACE (OBJ REGION) OF OBJECT WITH (CREATE REGION
                                                       WIDTH _ (FETCH (IMAGEBOX XSIZE) OF IMAGEBOX)
                                                       HEIGHT _ (FETCH (IMAGEBOX YSIZE) OF IMAGEBOX)))
          (REPLACE (OBJ YDESC) OF OBJECT WITH (FETCH (IMAGEBOX YDESC) OF IMAGEBOX))
          (REPLACE (OBJ XKERN) OF OBJECT WITH (FETCH (IMAGEBOX XKERN) OF IMAGEBOX])
```

(OBJ.COMPUTE.REGION

```
  [LAMBDA (WINDOW OBJECT)                                   (* bbb "11-Dec-85 14:29")
    (replace (REGION LEFT) of (fetch (OBJ REGION) of OBJECT) with (DSPXPOSITION NIL WINDOW))
    [replace (REGION BOTTOM) of (fetch (OBJ REGION) of OBJECT) with (ADD1 (IDIFFERENCE (DSPYPOSITION NIL WINDOW)
                                                                                       (fetch (OBJ YDESC) of OBJECT]
    (replace INSTANTIATED of OBJECT with T])
```

(OBJ.COPYBUTTONEVENTFN

```
  [LAMBDA (WINDOW)                                          (* jtm%: " 3-Nov-87 17:12")
                                                            (* rmk%: "16-May-86 14:48")

          (* Tracks the mouse, while the button is down objects are inverted and when the button is released either the user's
          COPYBUTTONEVENTFN is called or else a COPYINSERT is performed.)

    (PROG ((CLIPPING.REGION (DSPCLIPPINGREGION NIL WINDOW))
           BUTTON OLDPOS NOW NEAR COPYBUTTONEVENTINFN NOW.IMAGEOBJ OLDX OLDY)
                                                            (* note which button is down.)
          (TOTOPW WINDOW)
          (COND
             ((LASTMOUSESTATE LEFT)
              (SETQ BUTTON 'LEFT))
             ((LASTMOUSESTATE MIDDLE)
              (SETQ BUTTON 'MIDDLE))
             (T                                             (* no button down, not interested.)
                (RETURN)))                                  (* get the region of this window.)
          (SETQ NEAR (OBJ.FIND.OBJECT WINDOW (LASTMOUSEX WINDOW)
                                      (LASTMOUSEY WINDOW)))
      FLIP
          (if NOW
              then (OBJ.FLIP.OBJECT NOW WINDOW))
          (if NEAR
              then (OBJ.FLIP.OBJECT NEAR WINDOW))
```

```
      (SETQ NOW NEAR)
LP                                                          (* wait for a button up or move out of region)
      (GETMOUSESTATE)
      (COND
         ((NOT (LASTMOUSESTATE (OR LEFT MIDDLE)))           (* button up, process it.)
          (if NOW
              then (OBJ.FLIP.OBJECT NOW WINDOW)
                   (SETQ NOW.IMAGEOBJ (fetch (OBJ OBJECT) of NOW))
                                                            (* NOW node has been selected.)
                   (SETQ COPYBUTTONEVENTINFN (IMAGEOBJPROP NOW.IMAGEOBJ 'COPYBUTTONEVENTINFN))
                   [RETURN (if COPYBUTTONEVENTINFN
                               then (SETQ OLDX (DSPXPOSITION NIL WINDOW))
                                    (SETQ OLDY (DSPYPOSITION NIL WINDOW))
                                    (MOVETO (IPLUS (fetch (OBJ XKERN) of NOW)
                                                   (fetch (REGION LEFT) of (fetch (OBJ REGION) of NOW)))
                                            (IPLUS (fetch (OBJ YDESC) of NOW)
                                                   (fetch (REGION BOTTOM) of (fetch (OBJ REGION) of NOW)))
                                            WINDOW)
                                    (OBJ.APPLY.USER.FN COPYBUTTONEVENTINFN NOW.IMAGEOBJ WINDOW
                                           (fetch (OBJ REGION) of NOW))
                                    (MOVETO OLDX OLDY WINDOW)
                             else (COPYINSERT (APPLY* (IMAGEOBJPROP NOW.IMAGEOBJ 'COPYFN)
                                                      NOW.IMAGEOBJ]
            else (RETURN)))
         ((NOT (INSIDEP CLIPPING.REGION (LASTMOUSEX WINDOW)
                        (LASTMOUSEY WINDOW)))               (* outside of region, return)
          (if NOW
              then (OBJ.FLIP.OBJECT NOW WINDOW))
          (RETURN))
         ([EQ NOW (SETQ NEAR (OBJ.FIND.OBJECT WINDOW (LASTMOUSEX WINDOW)
                                              (LASTMOUSEY WINDOW]
          (GO LP))
         (T (GO FLIP])
```

(OBJ.DELFROMW.HORIZONTAL

```
  [LAMBDA (HWINDOW OBJECT)                                  ; Edited 12-Aug-93 23:01 by rmk:
                                                            (* bbb " 7-Jan-86 16:54")
```

;;; The object is deleted from HWINDOW, close up the display by readjusting the lefts of all the following objects--and then redisplay from the left of the ;;; deleted object to the right of the clipping region

```
    (LET*
     ((CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
      (CLIP.LEFT (FETCH (REGION LEFT) OF CLIPPING.REGION))
      (CLIP.RIGHT (FETCH (REGION RIGHT) OF CLIPPING.REGION))
      (CLIP.WIDTH (FETCH (REGION WIDTH) OF CLIPPING.REGION))
      (OBJECTS (WINDOWPROP HWINDOW 'OBJECTS))
      DELETED.OBJECT REGION.OF.DELETED.OBJECT LEFT.OF.DELETED.OBJECT RIGHT.OF.DELETED.OBJECT
      WIDTH.OF.DELETED.OBJECT OBJECTS.FOLLOWING WIDTH.OF.OBJECTS.FOLLOWING VISIBLE.WIDTH SCREEN.REDISPLAYED)
     [COND
        ((NULL OBJECTS)
         (ERROR "Object not found " OBJECT))
        ((EQ OBJECT (FETCH (OBJ OBJECT) OF (CAR OBJECTS)))
         (SETQ DELETED.OBJECT (CAR OBJECTS))
         (WINDOWPROP HWINDOW 'OBJECTS (CDR OBJECTS))
         (SETQ OBJECTS.FOLLOWING (CDR OBJECTS))
         (DSPXPOSITION 0 HWINDOW))
        (T (FOR OBJECTTAIL ON OBJECTS WHEN (EQ OBJECT (FETCH (OBJ OBJECT) OF (CADR OBJECTTAIL)))
              DO (SETQ DELETED.OBJECT (CADR OBJECTTAIL))
                 (IF (FETCH (OBJ INSTANTIATED) OF DELETED.OBJECT)
                     THEN (DSPXPOSITION (OBJ.END.OF.OBJECT HWINDOW (CAR OBJECTTAIL))
                                        HWINDOW))
                 (RPLACD OBJECTTAIL (CDDR OBJECTTAIL))
                 (SETQ OBJECTS.FOLLOWING (CDR OBJECTTAIL))
                 (RETURN)
              FINALLY (ERROR "Object not found " OBJECT]
     [IF (FETCH (OBJ INSTANTIATED) OF DELETED.OBJECT)
         THEN
         (SETQ REGION.OF.DELETED.OBJECT (FETCH (OBJ REGION) OF DELETED.OBJECT))
         (SETQ LEFT.OF.DELETED.OBJECT (FETCH (REGION LEFT) OF REGION.OF.DELETED.OBJECT))
         (SETQ RIGHT.OF.DELETED.OBJECT (FETCH (REGION RIGHT) OF REGION.OF.DELETED.OBJECT))
         (SETQ WIDTH.OF.DELETED.OBJECT (FETCH (REGION WIDTH) OF REGION.OF.DELETED.OBJECT))
                                                            ; If the deleted object was instantiated we will have to alter other
                                                            ; objects regions
         (FOR OBJECT IN OBJECTS.FOLLOWING WHEN (OR (FETCH (OBJ INSTANTIATED) OF OBJECT)
                                                   (ILESSP (DSPXPOSITION NIL HWINDOW)
                                                           CLIP.RIGHT))
            DO (IF (FETCH (OBJ INSTANTIATED) OF OBJECT)
                   THEN (REPLACE (REGION LEFT) OF (FETCH (OBJ REGION) OF OBJECT)
                           WITH (IDIFFERENCE (FETCH (REGION LEFT) OF (FETCH (OBJ REGION) OF OBJECT))
                                             WIDTH.OF.DELETED.OBJECT))
                 ELSE (OBJ.INSTANTIATE HWINDOW OBJECT))
               (DSPXPOSITION (OBJ.END.OF.OBJECT HWINDOW OBJECT)
                             HWINDOW))
         (IF (ILESSP (OBJ.END.OF.OBJECT HWINDOW DELETED.OBJECT)
                     CLIP.LEFT)
```

```
        THEN                                                    ; Object entirely to the left of clipping region so don't adjust
                                                                ; clipping region
             (WXOFFSET WIDTH.OF.DELETED.OBJECT HWINDOW)
             (OBJ.RECOMPUTE.EXTENT HWINDOW)
      ELSE ;; Move to the left the objects following and if these can't fill the clipping region move the object before back (if there is an object
           ;; before)
           (SETQ VISIBLE.WIDTH (ADD1 (IDIFFERENCE CLIP.RIGHT LEFT.OF.DELETED.OBJECT)))
           [SETQ WIDTH.OF.OBJECTS.FOLLOWING (FOR OBJECT IN OBJECTS.FOLLOWING UNTIL (GREATERP $$VAL
                                                                                          VISIBLE.WIDTH)
                                                    SUM (FETCH (REGION WIDTH) OF (FETCH (OBJ REGION)
                                                                                         OF OBJECT]
           (IF (ILESSP LEFT.OF.DELETED.OBJECT CLIP.LEFT)
               THEN                                             ; Object is partially to the left of the clipping region.
                    (WXOFFSET (IDIFFERENCE (FETCH (REGION LEFT) OF (FETCH (OBJ REGION)
                                                                          OF (CAR OBJECTS.FOLLOWING)))
                                           CLIP.LEFT)
                              HWINDOW)
                    (OBJ.RECOMPUTE.EXTENT HWINDOW)
                    (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
                    (REDISPLAYW HWINDOW CLIPPING.REGION T)
             ELSE (IF (ILESSP WIDTH.OF.OBJECTS.FOLLOWING VISIBLE.WIDTH)
                      THEN (WXOFFSET (IDIFFERENCE WIDTH.OF.DELETED.OBJECT WIDTH.OF.OBJECTS.FOLLOWING)
                                  HWINDOW)
                           (OBJ.RECOMPUTE.EXTENT HWINDOW)
                           (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
                           (REDISPLAYW HWINDOW CLIPPING.REGION T)
                    ELSE (OBJ.RECOMPUTE.EXTENT HWINDOW)
                         (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
                         (IF (REGIONSINTERSECTP REGION.OF.DELETED.OBJECT CLIPPING.REGION)
                             THEN (REDISPLAYW HWINDOW (CREATE REGION
                                                              USING CLIPPING.REGION WIDTH _
                                                                    (ADD1 (IDIFFERENCE (FETCH (REGION RIGHT)
                                                                                              OF CLIPPING.REGION)
                                                                                       LEFT.OF.DELETED.OBJECT))
                                                                    LEFT _ LEFT.OF.DELETED.OBJECT)
                                                      T]
     (IF (NULL (WINDOWPROP HWINDOW 'OBJECTS))
         THEN (OBJ.CLEARW HWINDOW))
     OBJECT])
```

(OBJ.DELFROMW.VERTICAL

```
  [LAMBDA (VWINDOW OBJECT)                                      ; Edited  3-Aug-93 09:28 by rmk:
                                                                ; (* bbb "20-Dec-85 14:25")
```

;;; The object is deleted from HWINDOW, close up the display by readjusting the tops of all the following objects--and then redisplay from the top of the ;;; deleted object to the bottom of the clipping region

```
    (LET* ((CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
           (CLIP.TOP (FETCH (REGION TOP) OF CLIPPING.REGION))
           (CLIP.HEIGHT (FETCH (REGION HEIGHT) OF CLIPPING.REGION))
           (OBJECTS (WINDOWPROP VWINDOW 'OBJECTS))
           DELETED.OBJECT REGION.OF.DELETED.OBJECT TOP.OF.DELETED.OBJECT HEIGHT.OF.DELETED.OBJECT
           OBJECTS.FOLLOWING SCREEN.REDISPLAYED)
          [COND
             ((NULL OBJECTS)
              (ERROR "Object not found " OBJECT))
             ((EQ OBJECT (FETCH (OBJ OBJECT) OF (CAR OBJECTS)))
              (SETQ DELETED.OBJECT (CAR OBJECTS))
              (WINDOWPROP VWINDOW 'OBJECTS (CDR OBJECTS))
              (SETQ OBJECTS.FOLLOWING (CDR OBJECTS))
              (DSPYPOSITION 0 VWINDOW))
             (T (FOR OBJECTTAIL ON OBJECTS WHEN (EQ OBJECT (FETCH (OBJ OBJECT) OF (CADR OBJECTTAIL)))
                   DO (SETQ DELETED.OBJECT (CADR OBJECTTAIL))
                      (IF (FETCH (OBJ INSTANTIATED) OF DELETED.OBJECT)
                          THEN (DSPYPOSITION (OBJ.END.OF.OBJECT VWINDOW (CAR OBJECTTAIL))
                                             VWINDOW))
                      (RPLACD OBJECTTAIL (CDDR OBJECTTAIL))
                      (SETQ OBJECTS.FOLLOWING (CDR OBJECTTAIL))
                      (RETURN)
                   FINALLY (ERROR "Object not found " OBJECT]
          [IF (FETCH (OBJ INSTANTIATED) OF DELETED.OBJECT)
              THEN (SETQ REGION.OF.DELETED.OBJECT (FETCH (OBJ REGION) OF DELETED.OBJECT))
                   (SETQ TOP.OF.DELETED.OBJECT (FETCH (REGION TOP) OF REGION.OF.DELETED.OBJECT))
                   (SETQ HEIGHT.OF.DELETED.OBJECT (FETCH (REGION HEIGHT) OF REGION.OF.DELETED.OBJECT))
                                                                ; If the deleted object was instantiated we will have to alter other
                                                                ; objects regions
                   (BIND (CLIP.BOTTOM _ (FETCH (REGION BOTTOM) OF CLIPPING.REGION)) FOR OBJECT IN
                                                                                          OBJECTS.FOLLOWING
                      UNTIL (AND (ILEQ (DSPYPOSITION NIL VWINDOW)
                                       CLIP.BOTTOM)
                                 (NOT (FETCH (OBJ INSTANTIATED) OF OBJECT)))
                      WHEN (OR (FETCH (OBJ INSTANTIATED) OF OBJECT)
                               (IGREATERP (DSPYPOSITION NIL VWINDOW)
                                          CLIP.BOTTOM))
                      DO (IF (FETCH (OBJ INSTANTIATED) OF OBJECT)
```

```
                                THEN (REPLACE (REGION BOTTOM) OF (FETCH (OBJ REGION) OF OBJECT)
                                        WITH (IPLUS (FETCH (REGION BOTTOM) OF (FETCH (OBJ REGION) OF OBJECT))
                                                HEIGHT.OF.DELETED.OBJECT))
                                ELSE (OBJ.INSTANTIATE VWINDOW OBJECT))
                              (DSPYPOSITION (OBJ.END.OF.OBJECT VWINDOW OBJECT)
                                    VWINDOW))
                        (IF (IGREATERP (OBJ.END.OF.OBJECT VWINDOW DELETED.OBJECT)
                                CLIP.TOP)
                            THEN                                    ; Object entirely to the top of clipping region so don't adjust
                                                                    ; clipping region
                                 (WYOFFSET (IMINUS HEIGHT.OF.DELETED.OBJECT)
                                           VWINDOW)
                                 (OBJ.RECOMPUTE.EXTENT VWINDOW)
                             ELSE (IF (IGREATERP TOP.OF.DELETED.OBJECT CLIP.TOP)
                                      THEN                          ; Object is partially in clipping region
                                           (IF (NOT OBJECTS.FOLLOWING)
                                               THEN

                                               ;; This is the very last object that we deleted.  We don't allow the user to scroll past the end of the
                                               ;; window so scroll back at most one screen

                                                   (IF (IGREATERP CLIP.TOP CLIP.HEIGHT)
                                                       THEN         ; WYOFFSET (PLUS EXISTING.OFFSET (IMINUS CLIP.TOP))
                                                                    ; VWINDOW
                                                     ELSE           ; WYOFFSET (PLUS EXISTING.OFFSET (IMINUS
                                                                    ; CLIP.HEIGHT)) VWINDOW
                                                       )
                                                   (OBJ.RECOMPUTE.EXTENT VWINDOW)
                                                   (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
                                                   (REDISPLAYW VWINDOW CLIPPING.REGION T)
                                                   (SETQ SCREEN.REDISPLAYED T)
                                               ELSE (WYOFFSET (IDIFFERENCE CLIP.TOP TOP.OF.DELETED.OBJECT)
                                                              VWINDOW)

                                                   ;; Adjust the amount we're looking at by the amount of the deleted object that wasn't in the
                                                   ;; clipping region

                                                   ))
                                  (IF (NOT SCREEN.REDISPLAYED)
                                      THEN (OBJ.RECOMPUTE.EXTENT VWINDOW)
                                           (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
                                           (IF (REGIONSINTERSECTP REGION.OF.DELETED.OBJECT CLIPPING.REGION)
                                               THEN (REDISPLAYW VWINDOW [CREATE REGION
                                                                          USING CLIPPING.REGION HEIGHT _
                                                                                (ADD1 (IDIFFERENCE
                                                                                       TOP.OF.DELETED.OBJECT
                                                                                       (FETCH (REGION BOTTOM)
                                                                                          OF CLIPPING.REGION]
                                                                                T]
                (IF (NULL (WINDOWPROP VWINDOW 'OBJECTS))
                    THEN (OBJ.CLEARW VWINDOW))
               OBJECT])
```

(OBJ.DRAW.OBJECT

```
  [LAMBDA (WINDOW OBJECT)                                           ; Edited 25-Nov-96 21:16 by rmk:
                                                                    (* bbb "12-Dec-85 12:29")
    (PROG ((OLDX (DSPXPOSITION NIL WINDOW))
           (OLDY (DSPYPOSITION NIL WINDOW)))
          (MOVETO (PLUS (FETCH (OBJ XKERN) OF OBJECT)
                        (FETCH (REGION LEFT) OF (FETCH (OBJ REGION) OF OBJECT)))
                  (PLUS (FETCH (REGION BOTTOM) OF (FETCH (OBJ REGION) OF OBJECT))
                        (FETCH (OBJ YDESC) OF OBJECT))
                  WINDOW)
          (IF (IMAGEOBJP (FETCH (OBJ OBJECT) OF OBJECT))
              THEN (APPLY* (IMAGEOBJPROP (FETCH (OBJ OBJECT) OF OBJECT)
                                         'DISPLAYFN)
                           (FETCH (OBJ OBJECT) OF OBJECT)
                           (GETSTREAM WINDOW))
            ELSE (APPLY* (WINDOWPROP WINDOW 'DISPLAYFN)
                         (FETCH (OBJ OBJECT) OF OBJECT)
                         (GETSTREAM WINDOW)))
          (IF (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
                  'HORIZONTAL)
              THEN (MOVETO (OBJ.END.OF.OBJECT WINDOW OBJECT)
                           OLDY WINDOW)
            ELSE (MOVETO OLDX (OBJ.END.OF.OBJECT WINDOW OBJECT)
                         WINDOW])
```

(OBJ.END.OF.OBJECT

```
  [LAMBDA (WINDOW OBJECT FLIPVERTICAL)                              ; Edited 25-Nov-96 21:16 by rmk:
                                                                    (* bbb "16-Dec-85 16:21")
    ;; Returns negative values for vertical window if FLIPVERTICAL.  This helps to unify horizontal and vertical calculations, compensating for the fact
    ;; that vertical positions are measured bottom-up, horizontal are measured left-right, and we want to draw objects left-right but top-down.

    (IF (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
            'HORIZONTAL)
        THEN (PLUS (FETCH (REGION LEFT) OF (FETCH (OBJ REGION) OF OBJECT))
```

```
                  (FETCH (REGION WIDTH) OF (FETCH (OBJ REGION) OF OBJECT))
                  (WINDOWPROP WINDOW 'SEPARATIONDISTANCE))
      ELSEIF FLIPVERTICAL
        THEN (DIFFERENCE (WINDOWPROP WINDOW 'SEPARATIONDISTANCE)
                  (FETCH (REGION BOTTOM) OF (FETCH (OBJ REGION) OF OBJECT)))
      ELSE (DIFFERENCE (FETCH (REGION BOTTOM) OF (FETCH (OBJ REGION) OF OBJECT))
                 (WINDOWPROP WINDOW 'SEPARATIONDISTANCE])
```

(OBJ.FIND.OBJECT

```
  [LAMBDA (WINDOW MOUSEX MOUSEY)                                    (* bbb "19-Dec-85 14:34")
    (LET [(OBJECT (if (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
                          'HORIZONTAL)
                      then (for OBJECT in (WINDOWPROP WINDOW 'OBJECTS)
                              thereis (AND (ILEQ (fetch (REGION LEFT) of (fetch (OBJ REGION) of OBJECT))
                                                 MOUSEX)
                                           (IGEQ (fetch (REGION RIGHT) of (fetch (OBJ REGION) of OBJECT))
                                                 MOUSEX))
                              repeatuntil (IGREATERP (OBJ.END.OF.OBJECT WINDOW OBJECT)
                                                     MOUSEX))
                    else (for OBJECT in (WINDOWPROP WINDOW 'OBJECTS)
                            thereis (AND (IGEQ (fetch (REGION TOP) of (fetch (OBJ REGION) of OBJECT))
                                               MOUSEY)
                                         (ILEQ (fetch (REGION BOTTOM) of (fetch (OBJ REGION) of OBJECT))
                                               MOUSEY))
                            repeatuntil (ILESSP (OBJ.END.OF.OBJECT WINDOW OBJECT)
                                                MOUSEY]
          OBJECT])
```

(OBJ.FIND.REGION.HORIZONTAL

```
  [LAMBDA (HWINDOW SEARCHOBJECT)                                    (* bbb "11-Dec-85 10:52")
```

(* The object SEARCHOBJECT is searched for and its region is returned.
This may involve instantiating objects.)

```
    (LET ((OLDX (DSPXPOSITION NIL HWINDOW))
          FOUND)
         (DSPXPOSITION [fetch (REGION LEFT) of (fetch (OBJ REGION) of (CAR (WINDOWPROP HWINDOW 'OBJECTS]
                HWINDOW)
         (for OBJECT in (WINDOWPROP HWINDOW 'OBJECTS)
            do (if (NOT (fetch (OBJ INSTANTIATED) of OBJECT))
                   then (if (EQ SEARCHOBJECT (fetch (OBJ OBJECT) of OBJECT))
                            then (SETQ FOUND T))
                        (OBJ.COMPUTE.IMAGEBOX HWINDOW OBJECT)
                        (OBJ.COMPUTE.REGION HWINDOW OBJECT)
                        (SETQ OLDX (OBJ.END.OF.OBJECT HWINDOW OBJECT))
                        (DSPXPOSITION OLDX HWINDOW)
                 else (DSPXPOSITION (OBJ.END.OF.OBJECT HWINDOW OBJECT)
                                    HWINDOW))
            repeatuntil (OR (EQ SEARCHOBJECT (fetch (OBJ OBJECT) of OBJECT))
                            FOUND)
            finally (DSPXPOSITION OLDX HWINDOW)
                    (WINDOWPROP HWINDOW 'OLDXPOSITION (DSPXPOSITION NIL HWINDOW))
                    (WINDOWPROP HWINDOW 'OLDYPOSITION (DSPYPOSITION NIL HWINDOW))
                    (if (OR (EQ SEARCHOBJECT (fetch (OBJ OBJECT) of OBJECT))
                            FOUND)
                        then (RETURN (fetch (OBJ REGION) of OBJECT])
```

(OBJ.FIND.REGION.VERTICAL

```
  [LAMBDA (VWINDOW SEARCHOBJECT)                                    (* bbb "12-Dec-85 14:07")
```

(* The object SEARCHOBJECT is searched for and its region is returned.
This may involve instantiating objects.)

```
    (LET ((OLDY (DSPYPOSITION NIL VWINDOW))
          FOUND)
         (DSPYPOSITION [fetch (REGION TOP) of (fetch (OBJ REGION) of (CAR (WINDOWPROP VWINDOW 'OBJECTS]
                VWINDOW)
         (for OBJECT in (WINDOWPROP VWINDOW 'OBJECTS)
            do (if (NOT (fetch (OBJ INSTANTIATED) of OBJECT))
                   then (if (EQ SEARCHOBJECT (fetch (OBJ OBJECT) of OBJECT))
                            then (SETQ FOUND T))
                        (OBJ.COMPUTE.IMAGEBOX VWINDOW OBJECT)
                        (RELMOVETO 0 (IMINUS (fetch (OBJ ASCENT) of OBJECT))
                                   VWINDOW)
                        (OBJ.COMPUTE.REGION VWINDOW OBJECT)
                        (SETQ OLDY (OBJ.END.OF.OBJECT VWINDOW OBJECT))
                        (DSPYPOSITION OLDY VWINDOW)
                 else (DSPYPOSITION (OBJ.END.OF.OBJECT VWINDOW OBJECT)
                                    VWINDOW))
            repeatuntil (OR (EQ SEARCHOBJECT (fetch (OBJ OBJECT) of OBJECT))
                            FOUND)
            finally (DSPYPOSITION OLDY VWINDOW)
                    (WINDOWPROP VWINDOW 'OLDXPOSITION (DSPXPOSITION NIL VWINDOW))
                    (WINDOWPROP VWINDOW 'OLDYPOSITION (DSPYPOSITION NIL VWINDOW))
```

```
                (if (OR (EQ SEARCHOBJECT (fetch (OBJ OBJECT) of OBJECT))
                        FOUND)
                    then (RETURN (fetch (OBJ REGION) of OBJECT])
```

(OBJ.FLIP.OBJECT

```
[LAMBDA (OBJECT WINDOW)                                    (* bbb "11-Dec-85 10:46")
  (LET ((REGION (fetch (OBJ REGION) of OBJECT)))
       (BLTSHADE BLACKSHADE WINDOW (fetch (REGION LEFT) of REGION)
              (fetch (REGION BOTTOM) of REGION)
              (fetch (REGION WIDTH) of REGION)
              (fetch (REGION HEIGHT) of REGION)
              'INVERT
              (DSPCLIPPINGREGION NIL WINDOW])
```

(OBJ.HARDCOPYFN

```
[LAMBDA (WINDOW STREAM)                                    ; Edited 27-Nov-96 10:33 by rmk:
  ;; First make sure that everything is instantiated
  (FOR OBJECT BOX TOP (FIRSTTIME _ T)
       [SEPDISTANCE _ (TIMES (DSPSCALE NIL STREAM)
                             (WINDOWPROP WINDOW 'SEPARATIONDISTANCE]
       (LMARG _ (DSPLEFTMARGIN NIL STREAM))
       (RMARG _ (DSPRIGHTMARGIN NIL STREAM))
       (BMARG _ (DSPBOTTOMMARGIN NIL STREAM))
       (WINDOWTYPE _ (WINDOWPROP WINDOW 'WINDOWTYPE)) IN (WINDOWPROP WINDOW 'OBJECTS)
    DO ;; First make sure that OBJECT is instantiated, as if we had scrolled over it
       (OBJ.INSTANTIATE WINDOW OBJECT)
       (SETQ OBJECT (FETCH (OBJ OBJECT) OF OBJECT))
       ;; Then compute the imagebox for this particular stream
       (SETQ BOX (APPLY* (IMAGEOBJPROP OBJECT 'IMAGEBOXFN)
                         OBJECT STREAM))
       ;; Finally display the thing
       (IF FIRSTTIME
           THEN (SETQ FIRSTTIME NIL)
         ELSEIF (IF (EQ WINDOWTYPE 'HORIZONTAL)
                    THEN (GREATERP (+ (DSPXPOSITION NIL STREAM)
                                      (FETCH XSIZE OF BOX))
                                   RMARG)
                  ELSE (LESSP (- (DSPYPOSITION NIL STREAM)
                                 (FETCH YSIZE OF BOX))
                              BMARG))
           THEN                                            ; Won't fit, go to new page
                (DSPNEWPAGE STREAM))
       (SETQ TOP (DSPYPOSITION NIL STREAM))
       (APPLY* (IMAGEOBJPROP OBJECT 'DISPLAYFN)
               OBJECT STREAM)
       (CL:IF (EQ WINDOWTYPE 'HORIZONTAL)
           (MOVETO (+ (DSPXPOSITION NIL STREAM)
                      SEPDISTANCE)
                   TOP STREAM)
           (MOVETO LMARG (- (DSPYPOSITION NIL STREAM)
                            SEPDISTANCE)
                   STREAM))])
```

(OBJ.INDEX.OBJECT

```
[LAMBDA (WINDOW XORYDELTA)                                 (* bbb "12-Dec-85 16:46")
  (LET* [(OBJECTS (WINDOWPROP WINDOW 'OBJECTS))
         (NOBJECTS (FLENGTH OBJECTS))
         (OBJPOS (FTIMES NOBJECTS XORYDELTA))
         (OBJNUM (FIX OBJPOS))
         (OBJREG (OBJ.FIND.REGION WINDOW (fetch (OBJ OBJECT) of (CAR (NTH OBJECTS (IMIN NOBJECTS (ADD1 OBJNUM]
```

(* Note%: although we do the check for the case where XORYDELTA = 1.0 we won't actually be able to scroll off the end of the object until we can add the window property about extent use in scrolling. This property is in Jazz but we may put it into Intermezzo LFG.)

```
        (if (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
                'HORIZONTAL)
            then (IPLUS (fetch (REGION LEFT) of OBJREG)
                        (FTIMES (if (FEQP XORYDELTA 1.0)
                                    then 1.0
                                  else (FDIFFERENCE OBJPOS OBJNUM))
                                (fetch (REGION WIDTH) of OBJREG)))
          else (IDIFFERENCE (fetch (REGION TOP) of OBJREG)
                            (FTIMES (if (FEQP XORYDELTA 1.0)
                                        then 1.0
                                      else (FDIFFERENCE OBJPOS OBJNUM))
                                    (fetch (REGION HEIGHT) of OBJREG])
```

(OBJ.INSTANTIATE

```
[LAMBDA (WINDOW OBJECT PREVOBJECT)                          ; Edited 25-Nov-96 20:53 by rmk:
                                                            (* bbb "19-Dec-85 11:46")
  (LET [(WINDOWTYPE (WINDOWPROP WINDOW 'WINDOWTYPE]
       (if (NOT (fetch (OBJ INSTANTIATED) of OBJECT))
           then (OBJ.COMPUTE.IMAGEBOX WINDOW OBJECT)
                (if PREVOBJECT
                    then (if (EQ WINDOWTYPE 'HORIZONTAL)
                             then (DSPXPOSITION (OBJ.END.OF.OBJECT WINDOW PREVOBJECT)
                                                WINDOW)
                             else (DSPYPOSITION (OBJ.END.OF.OBJECT WINDOW PREVOBJECT)
                                                WINDOW)))
                (if (EQ WINDOWTYPE 'VERTICAL)
                    then (RELMOVETO 0 (IMINUS (fetch (OBJ ASCENT) of OBJECT))
                                    WINDOW))
                (OBJ.COMPUTE.REGION WINDOW OBJECT))
       (if (EQ WINDOWTYPE 'HORIZONTAL)
           then (DSPXPOSITION (OBJ.END.OF.OBJECT WINDOW OBJECT)
                              WINDOW)
           else (DSPYPOSITION (OBJ.END.OF.OBJECT WINDOW OBJECT)
                              WINDOW])
```

(OBJ.MOVETO.LAST.INSTANTIATED.OBJECT

```
[LAMBDA (WINDOW OBJECTS)                                    (* bbb "19-Dec-85 13:58")
  (for OBJECTTAIL on OBJECTS unless (AND (CADR OBJECTTAIL)
                                         (fetch (OBJ INSTANTIATED) of (CADR OBJECTTAIL)))
     bind NEW.XORY do (SETQ NEW.XORY (OBJ.END.OF.OBJECT WINDOW (CAR OBJECTTAIL)))
                      (if (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
                              'HORIZONTAL)
                          then (DSPXPOSITION NEW.XORY WINDOW)
                          else (DSPYPOSITION NEW.XORY WINDOW))
                      (RETURN])
```

(OBJ.RECOMPUTE.EXTENT

```
[LAMBDA (WINDOW)                                            ; Edited  3-May-94 10:34 by rmk:
                                                            (* bbb "10-Dec-85 11:20")
```

;; Fakes up the EXTENT property so that the thumb-scrolling scale will be in terms of number of objects, not their actual widths. This gives ;; reasonable behavior, even if we haven't instantiated all the objects, hence don't know how wide they are. And of course, a scale in terms of true ;; widths wouldn't help the user, because HE has no idea until he's seen them all!

```
(PROG ((CLIPREG (DSPCLIPPINGREGION NIL WINDOW))
       [HORIZONTAL (EQ 'HORIZONTAL (WINDOWPROP WINDOW 'WINDOWTYPE]
       (OBJECTS (WINDOWPROP WINDOW 'OBJECTS))
       CLIPSTART CLIPEND NOBJECTS NUMBER.IN.CLIPPING.REGION NUMBER.PRIOR.TO.CLIPPING.REGION LAST.OBJ.FRACT
       STARTTAIL REGIONSIZE)
      (CL:UNLESS OBJECTS
          (OBJ.CLEAR.EXTENT WINDOW)
          (RETURN))
      [IF HORIZONTAL
          THEN (SETQ CLIPSTART (FETCH (REGION LEFT) OF CLIPREG))
               (SETQ CLIPEND (FETCH (REGION RIGHT) OF CLIPREG))
        ELSE ;; Flip vertical coordinates to compenstate for the fact that ypositions are measured top-down. OBJ.END.OF.OBJ will also return
             ;; negative values in the vertical case. But we still have to compensate below when looking at object regions.
             (SETQ CLIPSTART (MINUS (FETCH (REGION TOP) OF CLIPREG)))
             (SETQ CLIPEND (MINUS (FETCH (REGION BOTTOM) OF CLIPREG]
      (SETQ NOBJECTS (LENGTH OBJECTS))
```

;; NUMBER.TO.LEFT.OF.CLIPPING.REGION are the ones that won't be shown. STARTTAIL has the first possibly visible one. Switches on ;; HORIZONTAL because Y positions go from bottom up but we are mapping top to left.

```
      [SETQ NUMBER.PRIOR.TO.CLIPPING.REGION
       (FOR OTAIL OREG ON OBJECTS EACHTIME (SETQ OREG (FETCH (OBJ REGION) OF (CAR OTAIL)))
          UNTIL (IGEQ (OBJ.END.OF.OBJECT WINDOW (CAR OTAIL)
                                         T)
                      CLIPSTART)
          SUM 1 FINALLY (IF OTAIL
                            THEN (SETQ STARTTAIL OTAIL)
                            ELSE ;; It seems like everything is prior to the clipping region, so declare that the last one isn't
                                 (SETQ STARTTAIL (LAST OBJECTS))
                                 (SETQ OREG (FETCH (OBJ REGION) OF (CAR STARTTAIL)))
                                 (ADD $$VAL -1))
                        ;; LAST.OBJ.FRACT is the fraction of the last object that will NOT be seen.
                        (SETQ LAST.OBJ.FRACT (IF OREG
                                                 THEN (CL:IF HORIZONTAL
                                                             (FQUOTIENT (IDIFFERENCE CLIPSTART
                                                                                     (FETCH (REGION LEFT)
                                                                                        OF OREG))
                                                                        (FETCH (REGION WIDTH) OF OREG))
                                                             (FQUOTIENT (IDIFFERENCE CLIPSTART
                                                                                     (MINUS (FETCH (REGION TOP)
                                                                                                   OF OREG)))
                                                                        (FETCH (REGION HEIGHT) OF OREG)))
                                                 ELSE 0.0))
                        (RETURN (FPLUS $$VAL LAST.OBJ.FRACT]
```

```
;; SETQ NUMBER.IN.CLIPPING.REGION (FPLUS (FDIFFERENCE 1.0 LEFTFRACT) (if (ILESSP (fetch (REGION RIGHT) of (fetch (OBJ
;; REGION) of (CAR LEFTTAIL))) CLIPRIGHT) then (for OBJECT in (CDR LEFTTAIL) until (IGEQ (fetch (REGION RIGHT) of (fetch (OBJ
;; REGION) of OBJECT)) CLIPRIGHT) sum 1 finally (if OBJECT then (RETURN (FPLUS $$VAL (FQUOTIENT (IDIFFERENCE CLIPRIGHT (fetch
;; (REGION LEFT) of (fetch (OBJ REGION) of OBJECT))) (fetch (REGION WIDTH) of (fetch (OBJ REGION) of OBJECT))))))) else 0.0))

;; NUMBER.IN.CLIPPING.REGION are the ones that will be seen

[SETQ NUMBER.IN.CLIPPING.REGION
 (IF (ILESSP (OBJ.END.OF.OBJECT WINDOW (CAR STARTTAIL)
                    T)
             CLIPEND)
     THEN ;; All of starting object is visible, so there may be more
          [FPLUS (FDIFFERENCE 1.0 LAST.OBJ.FRACT)
                 (FOR OBJECT OREG IN (CDR STARTTAIL) WHILE (FETCH INSTANTIATED OF OBJECT)
                    UNTIL (IGEQ (OBJ.END.OF.OBJECT WINDOW OBJECT T)
                                CLIPEND)
                    SUM 1
                    FINALLY
                          ;; Add on the fact of the last object that is visible
                          (CL:WHEN OBJECT
                              (SETQ OREG (FETCH (OBJ REGION) OF OBJECT))
                              [RETURN (FPLUS $$VAL (CL:IF HORIZONTAL
                                                          (FQUOTIENT (IDIFFERENCE CLIPEND
                                                                                  (FETCH (REGION LEFT)
                                                                                         OF OREG))
                                                                     (FETCH (REGION WIDTH) OF OREG))
                                                          (FQUOTIENT (IDIFFERENCE
                                                                      CLIPEND
                                                                      (MINUS (FETCH (REGION TOP)
                                                                                    OF OREG)))
                                                                     (FETCH (REGION HEIGHT) OF OREG)))])]
     ELSE ;; Starting object ends in clipping region
          (CL:IF HORIZONTAL
              [FQUOTIENT [IDIFFERENCE CLIPEND (FETCH (REGION LEFT) OF (FETCH (OBJ REGION)
                                                                             OF (CAR STARTTAIL]
                         (FETCH (REGION WIDTH) OF (FETCH (OBJ REGION) OF (CAR STARTTAIL]
              [FQUOTIENT [IDIFFERENCE CLIPEND (MINUS (FETCH (REGION TOP) OF (FETCH (OBJ REGION)
                                                                                   OF (CAR STARTTAIL]
                         (FETCH (REGION HEIGHT) OF (FETCH (OBJ REGION) OF (CAR STARTTAIL])]

;; REGIONSIZE is computed by first calculating the total width (in points) if each object were as wide as the clipping region, and dividing that by
;; the number that will actually appear.  Thus, it estimates how big a fictional clippring region would have to be if the actual region were to contain
;; one average-size object.

(SETQ REGIONSIZE (FIX (FQUOTIENT (FTIMES NOBJECTS (CL:IF HORIZONTAL
                                                         (FETCH (REGION WIDTH) OF CLIPREG)
                                                         (FETCH (REGION HEIGHT) OF CLIPREG)))
                                  NUMBER.IN.CLIPPING.REGION)))

;; We now compute the start of the extent (left or bottom) by positioning so that the right number of items will (fictionally) be prior to CLIPSTART.

(WINDOWPROP WINDOW 'EXTENT (CL:IF HORIZONTAL
                               (CREATE REGION
                                       WIDTH _ REGIONSIZE
                                       LEFT _ (IDIFFERENCE CLIPSTART (TIMES REGIONSIZE (FQUOTIENT
                                                                                          NUMBER.PRIOR.TO.CLIPPING.REGION
                                                                                                NOBJECTS)
                                                                                          ))
                                       BOTTOM _ -1
                                       HEIGHT _ -1)
                               (CREATE REGION
                                       WIDTH _ -1
                                       LEFT _ -1
                                       BOTTOM _ (IDIFFERENCE [IDIFFERENCE CLIPSTART
                                                                          (MINUS (TIMES REGIONSIZE
                                                                                        (FQUOTIENT
                                                                                           NUMBER.PRIOR.TO.CLIPPING.REGION
                                                                                                 NOBJECTS]
                                                             REGIONSIZE)
                                       HEIGHT _ REGIONSIZE))])
```

(OBJ.REPAINTFN

```
[LAMBDA (WINDOW REGION)                                    (* bbb "22-Aug-86 17:21")

        (* * Go through and figure out what objects intersect with this region and redraw them)

  (LET ((OLDX (DSPXPOSITION NIL WINDOW))
        (OLDY (DSPYPOSITION NIL WINDOW))
        (WINDOWTYPE (WINDOWPROP WINDOW 'WINDOWTYPE))
       FIRST.OBJECT)
      [IF (WINDOWPROP WINDOW 'OBJECTS)
          THEN                                             (* Old code (SETQ FIRST.OBJECT
                                                           (CAR (WINDOWPROP WINDOW
                                                           (QUOTE OBJECTS)))) (MOVETO
                                                           (fetch (REGION LEFT) of (fetch (OBJ REGION) of
                                                           FIRST.OBJECT)) (PLUS (fetch (REGION BOTTOM) of
```

```
                                                                (fetch (OBJ REGION) of FIRST.OBJECT))
                                                                (fetch (OBJ YDESC) of FIRST.OBJECT)) WINDOW))
            (OBJ.MOVETO.LAST.INSTANTIATED.OBJECT WINDOW (WINDOWPROP WINDOW 'OBJECTS]
        (bind IMAGEBOX for OBJECT in (WINDOWPROP WINDOW 'OBJECTS)
           do [if (NOT (fetch INSTANTIATED of OBJECT))
                  then (OBJ.COMPUTE.IMAGEBOX WINDOW OBJECT)
                       (if (EQ WINDOWTYPE 'VERTICAL)
                           then (RELMOVETO 0 (IMINUS (fetch (OBJ ASCENT) of OBJECT))
                                           WINDOW))
                       (OBJ.COMPUTE.REGION WINDOW OBJECT)
                       (if (EQ WINDOWTYPE 'HORIZONTAL)
                           then (SETQ OLDX (OBJ.END.OF.OBJECT WINDOW OBJECT))
                         else (SETQ OLDY (OBJ.END.OF.OBJECT WINDOW OBJECT]
              (if (REGIONSINTERSECTP (fetch (OBJ REGION) of OBJECT)
                                     REGION)
                  then (OBJ.DRAW.OBJECT WINDOW OBJECT))
              (if [OR (AND (EQ WINDOWTYPE 'HORIZONTAL)
                           (IGEQ (OBJ.END.OF.OBJECT WINDOW OBJECT)
                                 (fetch (REGION RIGHT) of REGION)))
                      (AND (EQ WINDOWTYPE 'VERTICAL)
                           (ILEQ (OBJ.END.OF.OBJECT WINDOW OBJECT)
                                 (fetch (REGION BOTTOM) of REGION]
                  then (RETURN)
                else (MOVETO OLDX OLDY WINDOW)))
        (MOVETO OLDX OLDY WINDOW)
        (WINDOWPROP WINDOW 'OLDXPOSITION (DSPXPOSITION NIL WINDOW))
        (WINDOWPROP WINDOW 'OLDYPOSITION (DSPYPOSITION NIL WINDOW])
```

(OBJ.REPLACE.HORIZONTAL

```
  [LAMBDA (HWINDOW OLD.OBJECT NEW.OBJECT DONT.REDISPLAY.FLG)      ; Edited 27-Jul-93 17:11 by rmk:
                                                                  (* bbb "19-Dec-85 16:40")
```

;;; Replaces new object with old object and adjusts the region of all objects to its left

```
    (LET* ((CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
           (CLIP.LEFT (FETCH (REGION LEFT) OF CLIPPING.REGION))
           (CLIP.RIGHT (FETCH (REGION RIGHT) OF CLIPPING.REGION))
           (OBJECTS (WINDOWPROP HWINDOW 'OBJECTS))
           OBJECTS.TAIL OBJ OLD.REGION NEW.REGION WIDTH.CHANGE WIDTH.SHOWING LEFT.OF.OLD.OBJECT
           END.OF.OLD.OBJECT)
          (FOR OBJECT ON OBJECTS WHEN (EQ OLD.OBJECT (FETCH (OBJ OBJECT) OF (CAR OBJECT)))
             DO (REPLACE (OBJ OBJECT) OF (CAR OBJECT) WITH NEW.OBJECT)
                (SETQ OBJ (CAR OBJECT))
                (SETQ END.OF.OLD.OBJECT (IF (FETCH (OBJ INSTANTIATED) OF OBJ)
                                            THEN (OBJ.END.OF.OBJECT HWINDOW OBJ)))
                (SETQ OLD.REGION (FETCH (OBJ REGION) OF (CAR OBJECT)))
                (SETQ OBJECTS.TAIL (CDR OBJECT))
                (RETURN)
             FINALLY (ERROR "Object not found " OLD.OBJECT))
          ;; Clear the screen starting at the replaced object, if necessary compute the new region and change the region of all following objects (if
          ;; they're instantiated) and redraw those that are on the screen.
          [IF (FETCH (OBJ INSTANTIATED) OF OBJ)
              THEN (SETQ LEFT.OF.OLD.OBJECT (FETCH (REGION LEFT) OF OLD.REGION))
                   (DSPXPOSITION LEFT.OF.OLD.OBJECT HWINDOW)
                   (REPLACE (OBJ INSTANTIATED) OF OBJ WITH NIL)
                   (OBJ.INSTANTIATE HWINDOW OBJ)
                   (SETQ NEW.REGION (FETCH (OBJ REGION) OF OBJ))
                   (DSPXPOSITION (OBJ.END.OF.OBJECT HWINDOW OBJ)
                                 HWINDOW)
                   (SETQ WIDTH.CHANGE (IDIFFERENCE (FETCH (REGION WIDTH) OF NEW.REGION)
                                                   (FETCH (REGION WIDTH) OF OLD.REGION]
          (FOR OBJECT IN OBJECTS.TAIL WHEN (OR (FETCH (OBJ INSTANTIATED) OF OBJECT)
                                               (ILESSP (DSPXPOSITION NIL HWINDOW)
                                                       CLIP.RIGHT))
             DO (IF (FETCH (OBJ INSTANTIATED) OF OBJECT)
                    THEN (REPLACE (REGION LEFT) OF (FETCH (OBJ REGION) OF OBJECT)
                                  WITH (IPLUS (FETCH (REGION LEFT) OF (FETCH (OBJ REGION) OF OBJECT))
                                              WIDTH.CHANGE))
                  ELSE (OBJ.INSTANTIATE HWINDOW OBJECT))
                (DSPXPOSITION (OBJ.END.OF.OBJECT HWINDOW OBJECT)
                              HWINDOW))
          (IF (AND (NULL DONT.REDISPLAY.FLG)
                   (FETCH (OBJ INSTANTIATED) OF OBJ))
              THEN (IF (ILESSP END.OF.OLD.OBJECT CLIP.LEFT)
                       THEN ;; Object is entirely to the left of the clipping region, adjust the clipping region but visually leave everything the
                            ;; same
                            (WXOFFSET (IMINUS WIDTH.CHANGE)
                                   HWINDOW)
                            (OBJ.RECOMPUTE.EXTENT HWINDOW)
                     ELSEIF (ILESSP LEFT.OF.OLD.OBJECT CLIP.LEFT)
                       THEN ;; Old object is partially in the clipping region.  In the case where the new object has a smaller area than the
                            ;; amount of the old object that is showing we align the new object at the left edge of the clipping region.
                            ;; Otherwise we will see the same amount of the new object as the old object
```

```
                        (SETQ WIDTH.SHOWING (ADD1 (IDIFFERENCE (FETCH (REGION RIGHT) OF OLD.REGION)
                                                               CLIP.LEFT)))
                        (IF (ILESSP (FETCH (REGION WIDTH) OF NEW.REGION)
                                    WIDTH.SHOWING)
                            THEN (WXOFFSET (IDIFFERENCE CLIP.LEFT LEFT.OF.OLD.OBJECT)
                                           HWINDOW)
                                 (OBJ.RECOMPUTE.EXTENT HWINDOW)
                                 (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
                                 (REDISPLAYW HWINDOW CLIPPING.REGION T)
                          ELSE (WXOFFSET (IMINUS WIDTH.CHANGE)
                                         HWINDOW)
                               (OBJ.RECOMPUTE.EXTENT HWINDOW)
                               (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
                               (REDISPLAYW HWINDOW (CREATE REGION USING CLIPPING.REGION WIDTH _
                                                                        WIDTH.SHOWING)
                                                   T))
                    ELSE (OBJ.RECOMPUTE.EXTENT HWINDOW)
                         (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL HWINDOW))
                         (IF (REGIONSINTERSECTP NEW.REGION CLIPPING.REGION)
                             THEN (REDISPLAYW HWINDOW [CREATE REGION USING CLIPPING.REGION LEFT _
                                                                           (FETCH (REGION LEFT) OF
                                                                                                NEW.REGION
                                                                           )
                                                                           WIDTH _
                                                                           (ADD1 (IDIFFERENCE
                                                                                  (FETCH (REGION RIGHT)
                                                                                     OF CLIPPING.REGION)
                                                                                  (FETCH (REGION LEFT)
                                                                                     OF NEW.REGION]
                                                  T])
```

(OBJ.REPLACE.VERTICAL

```
[LAMBDA (VWINDOW OLD.OBJECT NEW.OBJECT DONT.REDISPLAY.FLG)      (* bbb "19-Dec-85 16:45")
```

(* * Replaces new object with old object and adjusts the region of all objects to its top)

```
  (LET* ((CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
         (CLIP.TOP (fetch (REGION TOP) of CLIPPING.REGION))
         (CLIP.BOTTOM (fetch (REGION BOTTOM) of CLIPPING.REGION))
         (OBJECTS (WINDOWPROP VWINDOW 'OBJECTS))
         OBJECTS.TAIL OBJ OLD.REGION NEW.REGION HEIGHT.CHANGE HEIGHT.SHOWING TOP.OF.OLD.OBJECT
         END.OF.OLD.OBJECT)
        (for OBJECT on OBJECTS when (EQ OLD.OBJECT (fetch (OBJ OBJECT) of (CAR OBJECT)))
           do (replace (OBJ OBJECT) of (CAR OBJECT) with NEW.OBJECT)
              (SETQ OBJ (CAR OBJECT))
              (SETQ END.OF.OLD.OBJECT (if (fetch (OBJ INSTANTIATED) of OBJ)
                                          then (OBJ.END.OF.OBJECT VWINDOW OBJ)))
              (SETQ OLD.REGION (fetch (OBJ REGION) of (CAR OBJECT)))
              (SETQ OBJECTS.TAIL (CDR OBJECT))
              (RETURN)
           finally (ERROR "Object not found " OLD.OBJECT))
```

(* Clear the screen starting at the replaced object, if necessary compute the new region and change the region of all following objects (if they're instantiated) and redraw those that are on the screen.)

```
        [if (fetch (OBJ INSTANTIATED) of OBJ)
            then (SETQ TOP.OF.OLD.OBJECT (fetch (REGION TOP) of OLD.REGION))
                 (DSPYPOSITION TOP.OF.OLD.OBJECT VWINDOW)
                 (replace (OBJ INSTANTIATED) of OBJ with NIL)
                 (OBJ.INSTANTIATE VWINDOW OBJ)
                 (SETQ NEW.REGION (fetch (OBJ REGION) of OBJ))
                 (DSPYPOSITION (OBJ.END.OF.OBJECT VWINDOW OBJ)
                        VWINDOW)
                 (SETQ HEIGHT.CHANGE (IDIFFERENCE (fetch (REGION HEIGHT) of NEW.REGION)
                                                  (fetch (REGION HEIGHT) of OLD.REGION]
        (for OBJECT in OBJECTS.TAIL when (OR (fetch (OBJ INSTANTIATED) of OBJECT)
                                             (IGREATERP (DSPYPOSITION NIL VWINDOW)
                                                        CLIP.BOTTOM))
           do (if (fetch (OBJ INSTANTIATED) of OBJECT)
                  then (replace (REGION BOTTOM) of (fetch (OBJ REGION) of OBJECT)
                          with (IDIFFERENCE (fetch (REGION BOTTOM) of (fetch (OBJ REGION) of OBJECT))
                                            HEIGHT.CHANGE))
                else (OBJ.INSTANTIATE VWINDOW OBJECT))
              (DSPYPOSITION (OBJ.END.OF.OBJECT VWINDOW OBJECT)
                     VWINDOW))
        (if (AND (NULL DONT.REDISPLAY.FLG)
                 (fetch (OBJ INSTANTIATED) of OBJ))
            then (if (IGREATERP END.OF.OLD.OBJECT CLIP.TOP)
                     then
```

(* Object is entirely to the top of the clipping region, adjust the clipping region but visually leave everything the same)

```
                          (WYOFFSET HEIGHT.CHANGE VWINDOW)
                          (OBJ.RECOMPUTE.EXTENT VWINDOW)
                    elseif (IGREATERP TOP.OF.OLD.OBJECT CLIP.TOP)
                      then
```

```
(* Old object is partially in the clipping region. In the case where the new object has a smaller area than the amount of the
old object that is showing we align the new object at the top edge of the clipping region.
Otherwise we will see the same amount of the new object as the old object)

                    [SETQ HEIGHT.SHOWING (ADD1 (IDIFFERENCE CLIP.TOP (fetch (REGION BOTTOM) of OLD.REGION]
                    (if (ILESSP (fetch (REGION HEIGHT) of NEW.REGION)
                                HEIGHT.SHOWING)
                        then (WYOFFSET (IDIFFERENCE TOP.OF.OLD.OBJECT CLIP.TOP)
                                       VWINDOW)
                             (OBJ.RECOMPUTE.EXTENT VWINDOW)
                             (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
                             (REDISPLAYW VWINDOW CLIPPING.REGION T)
                      else (WYOFFSET HEIGHT.CHANGE VWINDOW)
                           (OBJ.RECOMPUTE.EXTENT VWINDOW)
                           (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
                           (REDISPLAYW VWINDOW (create REGION using CLIPPING.REGION BOTTOM _
                                                                    (fetch (REGION BOTTOM) of NEW.REGION)
                                                                    HEIGHT _ HEIGHT.SHOWING)
                                       T))
               else (OBJ.RECOMPUTE.EXTENT VWINDOW)
                    (SETQ CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
                    (if (REGIONSINTERSECTP NEW.REGION CLIPPING.REGION)
                        then (REDISPLAYW VWINDOW (create REGION using CLIPPING.REGION BOTTOM _
                                                                      (fetch (REGION BOTTOM) of CLIPPING.REGION
                                                                             )
                                                                      HEIGHT _ (IDIFFERENCE
                                                                                 (fetch (REGION TOP)
                                                                                    of NEW.REGION)
                                                                                 (fetch (REGION BOTTOM)
                                                                                    of CLIPPING.REGION)))
                                             T])
```

(OBJ.RESHAPEFN

```
[LAMBDA (WINDOW OLDIMAGE IMAGEREGION OLDSCREENREGION)          ; Edited 28-Apr-95 16:12 by rmk:
                                                               (* bbb "13-May-86 15:18")
   (WINDOWPROP WINDOW 'EXTENT NIL)
   ;; The extent of an OBJ window is funny and confuses the RESHAPEBYREPAINTFN.  So we eliminated it first, then recompute it.
   (RESHAPEBYREPAINTFN WINDOW OLDIMAGE IMAGEREGION OLDSCREENREGION)
   (DSPRIGHTMARGIN 65535 WINDOW)
   (OBJ.RECOMPUTE.EXTENT WINDOW])
```

(OBJ.SCROLLFN

```
[LAMBDA (WINDOW XDELTA YDELTA CONTINUOUSFLG)                   ; Edited 21-Mar-95 16:00 by rmk:
                                                               (* bbb "11-Dec-85 10:49")
   (IF (WINDOWPROP WINDOW 'OBJECTS)
       THEN (IF (EQ (WINDOWPROP WINDOW 'WINDOWTYPE)
                    'HORIZONTAL)
                THEN (OBJ.SCROLLFN.HORIZONTAL WINDOW XDELTA YDELTA CONTINUOUSFLG)
              ELSE (OBJ.SCROLLFN.VERTICAL WINDOW XDELTA YDELTA CONTINUOUSFLG])
```

(OBJ.SCROLLFN.HORIZONTAL

```
[LAMBDA (HWINDOW XDELTA YDELTA CONTINUOUSFLG)                  ; Edited 21-Mar-95 16:00 by rmk:
                                                               (* bbb "14-May-86 17:00")
   (LET* [(OBJECTS (WINDOWPROP HWINDOW 'OBJECTS))
          (REGIONOFFIRST (FETCH (OBJ REGION) OF (CAR OBJECTS)))
          (LEFTOFFIRST (FETCH (REGION LEFT) OF (FETCH (OBJ REGION) OF (CAR OBJECTS]
         (IF (NOT (FLOATP YDELTA))
             THEN                                              ; Disallow thumb scrolling in the vertical direction because the
                                                               ; extent isn't defined
                  [IF (FLOATP XDELTA)
                      THEN (SETQ XDELTA (IDIFFERENCE (FETCH (REGION LEFT) OF (DSPCLIPPINGREGION NIL HWINDOW))
                                                     (OBJ.INDEX.OBJECT HWINDOW XDELTA]
                  ;; Make sure that all objects that will be shown are instantiated, so we can compute a valid true region
                  (FOR OBJECT PREV (NEWCLIPRIGHT _ (IDIFFERENCE (FETCH (REGION RIGHT) OF (DSPCLIPPINGREGION
                                                                                           NIL HWINDOW))
                                                                XDELTA))
                     IN OBJECTS DO (OBJ.INSTANTIATE HWINDOW OBJECT PREV)
                                   (IF (IGREATERP (OBJ.END.OF.OBJECT HWINDOW OBJECT)
                                                  NEWCLIPRIGHT)
                                       THEN (RETURN))
                                   (SETQ PREV OBJECT))
                  (OBJ.MOVETO.LAST.INSTANTIATED.OBJECT HWINDOW OBJECTS)
                  ;; We don't want to be limited by the fako extent during actual scrolling.  The fako extent is reset below, and its only purpose is
                  ;; to influence what shows up in the scroll bar.  The true 'right' is the right of the last instantiated object
                  (WINDOWPROP HWINDOW 'EXTENT
                         (CREATE REGION SMASHING (WINDOWPROP HWINDOW 'EXTENT)
                                                 LEFT _ LEFTOFFIRST WIDTH _
                                                 (IDIFFERENCE
                                                  [ADD1 (FOR OBJECT PREV IN OBJECTS
                                                           WHILE (FETCH (OBJ INSTANTIATED) OF OBJECT)
```

```
                                                    DO (SETQ PREV OBJECT)
                                                    FINALLY (RETURN (FETCH (REGION RIGHT)
                                                                           OF (FETCH (OBJ REGION)
                                                                                     OF PREV]
                                              LEFTOFFIRST)))
                    (SCROLLBYREPAINTFN HWINDOW XDELTA YDELTA CONTINUOUSFLG)
                    (OBJ.RECOMPUTE.EXTENT HWINDOW])
```

(OBJ.SCROLLFN.VERTICAL

```
  [LAMBDA (VWINDOW XDELTA YDELTA CONTINUOUSFLG)                    ; Edited 21-Mar-95 15:58 by rmk:
                                                                   (* bbb "14-May-86 17:03")
    (LET* ((OBJECTS (WINDOWPROP VWINDOW 'OBJECTS))
           (CLIPPING.REGION (DSPCLIPPINGREGION NIL VWINDOW))
           [TOPOFFIRST (FETCH (REGION TOP) OF (FETCH (OBJ REGION) OF (CAR OBJECTS]
           FAKO.HEIGHT)
          (IF (NOT (FLOATP XDELTA))
              THEN                                                 ; Disallow thumb scrolling in the x direction
                   [IF (FLOATP YDELTA)
                       THEN (SETQ YDELTA (IDIFFERENCE (FETCH (REGION TOP) OF CLIPPING.REGION)
                                                      (OBJ.INDEX.OBJECT VWINDOW YDELTA]

                   ;; Make sure that all objects that will be shown are instantiated, so we can compute a valid true region

                   (FOR OBJECT PREV (NEWCLIPBOTTOM _ (IDIFFERENCE (FETCH (REGION BOTTOM) OF CLIPPING.REGION)
                                                                  YDELTA))
                      IN OBJECTS DO (OBJ.INSTANTIATE VWINDOW OBJECT PREV)
                                    (IF (ILESSP (OBJ.END.OF.OBJECT VWINDOW OBJECT)
                                                NEWCLIPBOTTOM)
                                        THEN (RETURN))
                                    (SETQ PREV OBJECT))

                   ;; We don't want to be limited by the fako extent during actual scrolling.  The fako extent is reset below, and its only purpose is
                   ;; to influence what shows up in the scroll bar.  The true 'bottom' is the bottom of the last instantiated object
                                                                   ; If we are looking at everything we should not scroll!
                   (OBJ.MOVETO.LAST.INSTANTIATED.OBJECT VWINDOW OBJECTS)
                   (IF [OR (AND (EQ YDELTA 0)
                                (NEQ XDELTA 0))
                           (FOR OBJECT IN OBJECTS THEREIS (NOT (AND (FETCH (OBJ INSTANTIATED) OF OBJECT)
                                                                    (ILEQ (FETCH (REGION TOP)
                                                                                 OF (FETCH (OBJ REGION)
                                                                                           OF OBJECT))
                                                                          (FETCH (REGION TOP) OF
                                                                                             CLIPPING.REGION
                                                                                 ))
                                                                    (IGEQ (FETCH (REGION BOTTOM)
                                                                                 OF (FETCH (OBJ REGION)
                                                                                           OF OBJECT))
                                                                          (FETCH (REGION BOTTOM) OF
                                                                                             CLIPPING.REGION
                                                                                 ]
                       THEN [SETQ FAKO.HEIGHT (ADD1 (IDIFFERENCE TOPOFFIRST
                                                                 (FOR OBJECT PREV IN OBJECTS
                                                                    WHILE (FETCH (OBJ INSTANTIATED) OF OBJECT)
                                                                    DO (SETQ PREV OBJECT)
                                                                    FINALLY (RETURN (FETCH (REGION BOTTOM)
                                                                                           OF (FETCH (OBJ REGION)
                                                                                                     OF PREV]
                            (WINDOWPROP VWINDOW 'EXTENT (CREATE REGION SMASHING (WINDOWPROP VWINDOW
                                                                                            'EXTENT)
                                                                                BOTTOM _ (ADD1 (DIFFERENCE
                                                                                                  TOPOFFIRST
                                                                                                  FAKO.HEIGHT
                                                                                                  ))
                                                                                HEIGHT _ FAKO.HEIGHT))
                            (SCROLLBYREPAINTFN VWINDOW XDELTA YDELTA CONTINUOUSFLG)
                            (OBJ.RECOMPUTE.EXTENT VWINDOW])

)

(AND (GETD 'MODERNWINDOW.SETUP)
     (MODERNWINDOW.SETUP (FUNCTION OBJ.BUTTONEVENTINFN)))
```

FUNCTION INDEX

RECORD INDEX