

File created: 11-Jun-90 16:20:36 {DSK}<usr>local>lde>lispcore>library>GRAPHZOOM.;2

changes to: (VARS GRAPHZOOMCOMS)

previous date: 6-Sep-85 08:52:00 {DSK}<usr>local>lde>lispcore>library>GRAPHZOOM.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1983, 1984, 1985, 1990 by Venue & Xerox Corporation. All rights reserved.

```
(RPAQQ GRAPHZOOMCOMS
  ((FILES GRAPHER)
   (RECORDS ZOOMGRAPH ZOOMGRAPHNODE)
   (FNS MAKE.ZOOM.GRAPH ORIG.NODE.OF.GRAPH SCALE.GRAPH.FONT SCALE.GRAPH.NODE SCALE.GRAPH.NODES SCALE.GRAPH
     RESET.GRAPH.EXTENT ZOOM.GRAPH.WINDOW ADJUST.EXTENT ZOOM.GRAPH.ADDLINKFN ZOOM.GRAPH.ADDNODEFN
     ZOOM.GRAPH.DELETELINKFN ZOOM.GRAPH.DELETENODEFN ZOOM.GRAPH.FONTCHANGEFN ZOOM.GRAPH.MOVENODEFN
     SHOWZOOMGRAPH ZOOM.TO.CENTER)
   (COMS (* general functions for scaling)
     (FNS ABSWXOFFSET ABSWYOFFSET SCALE.REGION UNSCALE.POSITION SCALE.POSITION WINDOW.SCALE))
   (COMS (FNS PRESS.DECREASING.FONT.LIST IP.DECREASING.FONT.LIST)
     (GLOBALVARS PRESS.DECREASING.FONT.LIST IP.DECREASING.FONT.LIST)
     (INITVARS PRESS.DECREASING.FONT.LIST IP.DECREASING.FONT.LIST))))

(FILESLoad GRAPHER)

(DECLARE%: EVAL@COMPILE

(RECORD ZOOMGRAPH (NODELIST DISPLAYGRAPH SG.MOVENODEFN SG.ADDNODEFN SG.DELETENODEFN SG.ADDLINKFN SG.DELETELINKFN)
)

(RECORD ZOOMGRAPHNODE (SG.POSITION SG.FONT SG.LABEL SG.USERDATA SG.))
)

(DEFINEQ

(MAKE.ZOOM.GRAPH
  [LAMBDA (GRAPH SCALE)
    (* rrb " 6-NOV-83 12:07")
    (* returns a graph that is a scaled version of GRAPH)
    (create GRAPH
      GRAPHNODES _ (SCALE.GRAPH.NODES (fetch (GRAPH GRAPHNODES) of GRAPH)
        SCALE)
      SIDESFLG _ (fetch (GRAPH SIDESFLG) of GRAPH)
      DIRECTEDFLG _ (fetch (GRAPH DIRECTEDFLG) of GRAPH)
      GRAPH.MOVENODEFN _ (FUNCTION ZOOM.GRAPH.MOVENODEFN)
      GRAPH.ADDNODEFN _ (FUNCTION ZOOM.GRAPH.ADDNODEFN)
      GRAPH.DELETENODEFN _ (FUNCTION ZOOM.GRAPH.DELETENODEFN)
      GRAPH.ADDLINKFN _ (FUNCTION ZOOM.GRAPH.ADDLINKFN)
      GRAPH.DELETELINKFN _ (FUNCTION ZOOM.GRAPH.DELETELINKFN)
      GRAPH.FONTCHANGEFN _ (FUNCTION ZOOM.GRAPH.FONTCHANGEFN))

(ORIG.NODE.OF.GRAPH
  [LAMBDA (NODE INGRAPH CORRESPGRAPH)
    (* rrb " 1-NOV-83 19:02")
    (* returns the node in CORRESPGRAPH corresponding to NODE
    in INGRAPH.)
    (bind (NODEID _ (fetch (GRAPHNODE NODEID) of NODE)) for INND in (fetch (GRAPH GRAPHNODES) of INGRAPH)
      as CORND in (fetch (GRAPH GRAPHNODES) of CORRESPGRAPH) when (EQ (fetch (GRAPHNODE NODEID) of INND)
        NODEID)
      do (RETURN CORND]))

(SCALE.GRAPH.FONT
  [LAMBDA (FONT SCALE)
    (* rrb " 1-NOV-83 18:23")
    (* returns the closest font for this scale.)

    (* "LABEL" is an approximation of the label string. A fixed one is used rather than the label of the node so that all labels in
    the same font will scale to the same font.)

    (SCALE.FONT (QUOTIENT (STRINGWIDTH "LABEL" FONT)
      SCALE)
      "LABEL"))

(SCALE.GRAPH.NODE
  [LAMBDA (NODE SCALE)
    (* rrb " 6-Sep-85 08:51")
    (* returns a node that has been scaled.)
    (* keeps the same id's so that the links don't have to change.)
    (* SCALE is the reciprocal of scaling done in SCALE/GRAPH)

    (* this used to be create copying but this fails in the case where the nodeid is a list structure because node checks are done
    with EQ. -
    rrb)
```

```
(create GRAPHNODE using NODE NODEPOSITION _ (SCALE.POSITION (fetch (GRAPHNODE NODEPOSITION) of NODE)
                                                    SCALE)
      NODEFONT _ (SCALE.GRAPH.FONT (fetch (GRAPHNODE NODEFONT) of NODE)
                                                    SCALE])
```

(SCALE.GRAPH.NODES

```
[LAMBDA (NODELIST SCALE)
```

```
(* rrb " 1-NOV-83 11:05")
(* scales a list of nodes)
```

```
(for NODE in NODELIST collect (SCALE.GRAPH.NODE NODE SCALE])
```

(SCALE.GRAPH

```
[LAMBDA (SGWINDOW)
```

```
(* rrb " 8-NOV-83 12:35")
```

```
(* takes the SKETCH.GRAPH in SGWINDOW and recomputes
```

```
it to its current scale)
```

```
(PROG [(SCALEDGRAPH (MAKE.ZOOM.GRAPH (WINDOWPROP SGWINDOW 'SKETCH.GRAPH)
                                     (WINDOWPROP SGWINDOW 'SCALE)
                                     (WINDOWPROP SGWINDOW 'GRAPH SCALEDGRAPH)
                                     (RESET.GRAPH.EXTENT SCALEDGRAPH SGWINDOW)
                                     (RETURN SCALEDGRAPH])
```

(RESET.GRAPH.EXTENT

```
[LAMBDA (GRAPH WINDOW)
```

```
(* sets the extent of the graph onto the extent window property)
```

```
(WINDOWPROP WINDOW 'EXTENT (GRAPHREGION GRAPH])
```

(ZOOM.GRAPH.WINDOW

```
[LAMBDA (ITEM MENU BUTTON)
```

```
(* rrb " 8-NOV-83 13:47")
```

```
(* zooms the main sketch graph window.)
```

```
(PROG ((MAINW (WINDOWPROP (WFROMMENU MENU)
                          'MAINWINDOW))
      (SMALLOUTFACTOR 1.1)
      (LARGEOUTFACTOR 1.8)
      SMALLINFACOR LARGEINFACOR)
```

```
(* factors are reciprocals so that IN followed by small OUT should return to the same place.)
```

```
(SETQ SMALLINFACOR (FQUOTIENT 1.0 SMALLOUTFACTOR))
(SETQ LARGEINFACOR (FQUOTIENT 1.0 LARGEOUTFACTOR)) (* set the SCALE and offsets)
(ZOOM.TO.CENTER MAINW (SELECTQ (CADR (CADR ITEM))
                              (IN (SELECTQ BUTTON
                                     (MIDDLE LARGEINFACOR)
                                     SMALLINFACOR))
                              (SELECTQ BUTTON
                                     (MIDDLE LARGEOUTFACTOR)
                                     SMALLOUTFACTOR))) (* rescale the graph)
```

```
(SCALE.GRAPH MAINW)
(ADJUST.EXTENT MAINW)
(REDISPLAYGRAPH MAINW])
```

(ADJUST.EXTENT

```
[LAMBDA (WINDOW)
```

```
(* rrb " 8-NOV-83 13:51")
```

```
(* adjust the offsets of WINDOW so that the visible region of the window is all extent.
If there is more visible region than extent, it centers the extent.)
```

```
(PROG ((EXTENT (WINDOWPROP WINDOW 'EXTENT))
      (REG (DSPCLIPPINGREGION NIL WINDOW))
      REGOFF REEXT EXTOFF EXTEXT)
(COND
  ((GREATERP (SETQ REEXT (fetch (REGION WIDTH) of REG))
    (SETQ EXTEXT (fetch (REGION WIDTH) of EXTENT)))
    (* center in X)
    (ABSWXOFFSET (DIFFERENCE (fetch (REGION LEFT) of EXTENT)
      (IQUOTIENT (IDIFFERENCE REEXT EXTEXT)
        2))
      WINDOW))
  ((GREATERP (SETQ EXTOFF (fetch (REGION LEFT) of EXTENT))
    (SETQ REGOFF (fetch (REGION LEFT) of REG))) (* move it to the left)
    (ABSWXOFFSET EXTOFF WINDOW))
  ((GREATERP (IPLUS REGOFF REEXT)
    (SETQ EXTOFF (IPLUS EXTOFF EXTEXT))) (* move it to the right)
    (ABSWXOFFSET (DIFFERENCE EXTOFF REEXT)
      WINDOW))
(COND
  ((GREATERP (SETQ REEXT (fetch (REGION HEIGHT) of REG))
    (SETQ EXTEXT (fetch (REGION HEIGHT) of EXTENT)))
    (* center in Y)
    (ABSWYOFFSET (DIFFERENCE (fetch (REGION BOTTOM) of EXTENT)
      (IQUOTIENT (IDIFFERENCE REEXT EXTEXT)
        2))
      WINDOW))
  ((GREATERP (SETQ EXTOFF (fetch (REGION BOTTOM) of EXTENT))
```

```

      (SETQ REGOFF (fetch (REGION BOTTOM) of REG)))
      (* move it up)
    (ABSWYOFFSET EXTOFF WINDOW))
  ((GREATERP (IPLUS REGOFF REGEXT)
    (SETQ EXTOFF (IPLUS EXTOFF EXTEXT)))
    (* move it down)
    (ABSWYOFFSET (DIFFERENCE EXTOFF REGEXT)
      WINDOW])

```

(ZOOM.GRAPH.ADDLINKFN

```

[LAMBDA (FROM TO GRAPH WINDOW)
  (* rrb " 6-NOV-83 12:08")
  (* the link adding function for a sketch graph.)

  (PROG [(ORGGRAPH (WINDOWPROP WINDOW 'SKETCH.GRAPH)
    (GRAPHADDLINK (ORIG.NODE.OF.GRAPH FROM GRAPH ORGGRAPH)
      (ORIG.NODE.OF.GRAPH TO GRAPH ORGGRAPH)
      ORGGRAPH WINDOW)])

```

(ZOOM.GRAPH.ADDNODEFN

```

[LAMBDA (GRAPH WINDOW)
  (* rrb " 1-NOV-83 17:46")
  (* the node adding function for a sketch graph.)

  (PROG (NEWNODE)
    (COND
      ((SETQ NEWNODE (GRAPHADDNODE (WINDOWPROP WINDOW 'SKETCH.GRAPH)
        WINDOW)))

    (* calls the graphs addnode function to create the node then scale it to the sketch window.)

    (RETURN (SCALE.GRAPH.NODE NEWNODE (WINDOWPROP WINDOW 'SCALE]))

```

(ZOOM.GRAPH.DELETELINKFN

```

[LAMBDA (FROM TO GRAPH WINDOW)
  (* rrb " 6-NOV-83 12:08")
  (* the link adding function for a sketch graph.)

  (PROG [(ORGGRAPH (WINDOWPROP WINDOW 'SKETCH.GRAPH)
    (GRAPHDELETELINK (ORIG.NODE.OF.GRAPH FROM GRAPH ORGGRAPH)
      (ORIG.NODE.OF.GRAPH TO GRAPH ORGGRAPH)
      ORGGRAPH WINDOW)])

```

(ZOOM.GRAPH.DELETENODEFN

```

[LAMBDA (NODE GRAPH WINDOW)
  (* rrb " 6-NOV-83 12:08")
  (* the node deleting function for a sketch graph.)

  (PROG [(ORGGRAPH (WINDOWPROP WINDOW 'SKETCH.GRAPH)
    (RETURN (GRAPHDELETENODE (ORIG.NODE.OF.GRAPH NODE GRAPH ORGGRAPH)
      ORGGRAPH WINDOW)])

```

(ZOOM.GRAPH.FONTCHANGEFN

```

[LAMBDA (HOW NODE GRAPH WINDOW)
  (* rrb " 6-NOV-83 12:08")
  (* the node deleting function for a sketch graph.)

  (PROG (NEWFONT ORIGNODE)
    [SETQ NEWFONT (NEXTSIZEFONT HOW (fetch (GRAPHNODE NODEFONT) of (SETQ ORIGNODE
      (ORIG.NODE.OF.GRAPH NODE GRAPH
        (WINDOWPROP WINDOW
          'SKETCH.GRAPH)
      (COND
        (NEWFONT (replace (GRAPHNODE NODEFONT) of ORIGNODE with NEWFONT)])

```

(ZOOM.GRAPH.MOVENODEFN

```

[LAMBDA (NODE NEWPOS GRAPH WINDOW)
  (* rrb " 6-NOV-83 12:08")

  (* the move function for a sketch graph. Moves the original node and calls its move fn if any.)

  (PROG [(ORGGRAPH (WINDOWPROP WINDOW 'SKETCH.GRAPH))
    (SCALE (WINDOWPROP WINDOW 'SCALE)
      (GRAPHMOVENODE (ORIG.NODE.OF.GRAPH NODE GRAPH ORGGRAPH)
        (UNSCALE.POSITION NEWPOS SCALE)
        ORGGRAPH WINDOW)])

```

(SHOWZOOMGRAPH

```

[LAMBDA (GRAPH WINDOW LEFTBUTTONFN MIDDLEBUTTONFN TOPJUSTIFYFLG ALLOWEDITFLG INITSCALE)
  (* edited%: "14-Feb-84 13:30")

  (* puts a zoomable graph in the given window, creating one if a window is not given.)

  (PROG (SKETCH.GRAPH (INITSCALE (OR INITSCALE 1.0)))
    (COND
      ((LISTP GRAPH)

        (* should be a GRAPHP check but since it is a list there is no easy test.)

        NIL)
      ((NULL GRAPH)
        (SETQ GRAPH (create GRAPH)))

```

```

(T (\ILLEGAL.ARG GRAPH)))
(SETQ SKETCH.GRAPH (MAKE.ZOOM.GRAPH GRAPH INITSCALE))
(* put a title on so that there will be a place to right button.)
(SETQ WINDOW (SIZE/GRAPH/WINDOW SKETCH.GRAPH (OR WINDOW (AND ALLOWEDITFLG ""))
TOPJUSTIFYFLG))
(bind MENU for ATW in (ATTACHEDWINDOWS WINDOW) when (AND (SETQ MENU (WINDOWPROP ATW 'MENU))
(EQ (fetch (MENU WHENSELECTEDFN)
of (CAR MENU))
(FUNCTION ZOOM.GRAPH.WINDOW)))
do (* a zoom menu is already attached to this window.)
(RETURN)
finally (ATTACHMENU (create MENU
ITEMS _ ' ((LARGER 'IN "increases the size of the graph elements.")
(smaller 'OUT "decreases the size of the graph elements"))
CENTERFLG _ T
WHENSELECTEDFN _ (FUNCTION ZOOM.GRAPH.WINDOW)
MENUROWS _ 1
MENUBORDERSIZE _ 1)
WINDOW
'TOP))
(WINDOWPROP WINDOW 'SKETCH.GRAPH GRAPH)
(WINDOWPROP WINDOW 'SCALE INITSCALE)
(WINDOWPROP WINDOW 'GRAPH SKETCH.GRAPH)
(WINDOWADDPROP WINDOW 'REPAINTFN (FUNCTION REDISPLAYGRAPH))
(WINDOWADDPROP WINDOW 'RESHAPEFN (FUNCTION RESHAPEBYREPAINTFN))
(WINDOWPROP WINDOW 'SCROLLFN (FUNCTION SCROLLBYREPAINTFN))
[COND
(ALLOWEDITFLG (* change the mode to invert so lines can be erased by being
redrawn.)
(DSPOPERATION 'INVERT WINDOW)
(WINDOWPROP WINDOW 'RIGHTBUTTONFN (FUNCTION GRAPHEDITEVENTFN]
(WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION APPLYTOSELECTEDNODE))
(WINDOWPROP WINDOW 'BROWSER/LEFTFN LEFTBUTTONFN)
(WINDOWPROP WINDOW 'BROWSER/MIDDLEFN MIDDLEBUTTONFN)
(OPENW WINDOW)
(REDISPLAYGRAPH WINDOW)
(RETURN WINDOW])

```

(ZOOM.TO.CENTER

[LAMBDA (WINDOW FACTOR)

(* rrb "6-NOV-83 11:46")

(* adjusts the SCALE window property and the offsets of WINDOW so that they correspond to zooming by FACTOR towards the center.)

```

(PROG ((OLDSCALE (WINDOW.SCALE WINDOW))
(REG (DSPCLIPPINGREGION NIL WINDOW))
NEWSCALE)
(WINDOWPROP WINDOW 'SCALE (SETQ NEWSCALE (FTIMES OLDSCALE FACTOR)))
(ABSWXOFFSET (FIX (FQUOTIENT [FTIMES OLDSCALE (FPLUS (fetch (REGION LEFT) of REG)
(FTIMES (fetch (REGION WIDTH) of REG)
(FQUOTIENT (FDIFFERENCE 1.0 FACTOR)
2]
NEWSCALE))
WINDOW)
(ABSWYOFFSET (FIX (FQUOTIENT [FTIMES OLDSCALE (FPLUS (fetch (REGION BOTTOM) of REG)
(FTIMES (fetch (REGION HEIGHT) of REG)
(FQUOTIENT (FDIFFERENCE 1.0 FACTOR)
2]
NEWSCALE))
WINDOW)
(* scale the EXTENT also.)
(AND (SETQ REG (WINDOWPROP WINDOW 'EXTENT))
(WINDOWPROP WINDOW 'EXTENT (SCALE.REGION REG FACTOR])
)

```

(* * general functions for scaling)

(DEFINEQ

(ABSWXOFFSET

[LAMBDA (NEWX W)

(* rrb "29-MAR-83 11:27")

(* sets the offset of a window.)

```

(WXOFFSET (IDIFFERENCE (WXOFFSET NIL W)
NEWX)
W])

```

(ABSWYOFFSET

[LAMBDA (NEWY W)

(* rrb "29-MAR-83 11:28")

(* sets the offset of a window.)

```

(WYOFFSET (IDIFFERENCE (WYOFFSET NIL W)
NEWY)
W])

```

(SCALE.REGION

[LAMBDA (REGION SCALE)

(* rrb "15-AUG-83 17:30")

(* scales a region into a windows coordinate space.)

```

(CREATEREGION (FIXR (QUOTIENT (fetch (REGION LEFT) of REGION)
                             SCALE))
  (FIXR (QUOTIENT (fetch (REGION BOTTOM) of REGION)
                 SCALE))
  (FIXR (QUOTIENT (fetch (REGION WIDTH) of REGION)
                 SCALE))
  (FIXR (QUOTIENT (fetch (REGION HEIGHT) of REGION)
                 SCALE]))

```

(UNSCALE.POSITION

[LAMBDA (POSITION SCALE)

(* rrb " 1-APR-83 16:05")

(* unscales a point in a window out into the larger coordinate space.)

```

(create POSITION
  XCOORD _ (TIMES (fetch (POSITION XCOORD) of POSITION)
                SCALE)
  YCOORD _ (TIMES (fetch (POSITION YCOORD) of POSITION)
                SCALE))

```

(SCALE.POSITION

[LAMBDA (POS SCALE)

(* rrb "29-APR-83 08:27")

(* scales a position from window coordinates into global coordinates.)

```

(create POSITION
  XCOORD _ (QUOTIENT (fetch (POSITION XCOORD) of POS)
                    SCALE)
  YCOORD _ (QUOTIENT (fetch (POSITION YCOORD) of POS)
                    SCALE))

```

(WINDOW.SCALE

[LAMBDA (SKETCHW)

(* rrb "14-MAR-83 10:31")

(* returns the scale of a sketch window.)

```

(WINDOWPROP SKETCHW 'SCALE))

```

)

(DEFINEQ

(PRESS.DECREASING.FONT.LIST

[LAMBDA NIL

(* rrb "10-Jun-85 14:35")

(* calculates, caches, and returns a list of the font descriptors for the fonts sketch windows are willing to print in.)

```

(OR (LISTP PRESS.DECREASING.FONT.LIST)
  (PROG (FONT)
    (RETURN (SETQ PRESS.DECREASING.FONT.LIST
      (APPEND (AND (SETQ FONT (FONTCREATE 'HELVETICA 36 NIL NIL NIL T))
                  (CONS FONT))
      (AND (SETQ FONT (FONTCREATE 'HELVETICAD 24 NIL NIL NIL T))
          (CONS FONT))
      (for SIZE in '(18 14 12 10 8 5 4 3) when (SETQ FONT (FONTCREATE 'HELVETICA SIZE NIL
                                                                    NIL NIL T))
        collect FONT]))

```

(IP.DECREASING.FONT.LIST

[LAMBDA NIL

(* rrb " 8-Jun-85 10:22")

(* * calculates, caches, returns a list of the font descriptors for the fonts sketch windows are willing to print in.)

(* this is calculated upon demand so that loading doesn't need fonts.)

```

(OR (LISTP IP.DECREASING.FONT.LIST)
  (SETQ IP.DECREASING.FONT.LIST (bind FONT for SIZE
    in '(36 30 24 18 14 12 10 8 6)
    when (SETQ FONT (FONTCREATE 'MODERN SIZE NIL NIL NIL T))
    join (CONS FONT))

```

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS PRESS.DECREASING.FONT.LIST IP.DECREASING.FONT.LIST)

)

(RPAQ? PRESS.DECREASING.FONT.LIST NIL)

(RPAQ? IP.DECREASING.FONT.LIST NIL)

(PUTPROPS GRAPHZOOM COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1990))

FUNCTION INDEX

ABSWXOFFSET	4	SCALE.GRAPH.FONT	1	ZOOM.GRAPH.ADDNODEFN	3
ABSWYOFFSET	4	SCALE.GRAPH.NODE	1	ZOOM.GRAPH.DELETELINKFN	3
ADJUST.EXTENT	2	SCALE.GRAPH.NODES	2	ZOOM.GRAPH.DELETENODEFN	3
IP.DECREASING.FONT.LIST	5	SCALE.POSITION	5	ZOOM.GRAPH.FONTCHANGEFN	3
MAKE.ZOOM.GRAPH	1	SCALE.REGION	5	ZOOM.GRAPH.MOVENODEFN	3
ORIG.NODE.OF.GRAPH	1	SHOWZOOMGRAPH	3	ZOOM.GRAPH.WINDOW	2
PRESS.DECREASING.FONT.LIST	5	UNSCALE.POSITION	5	ZOOM.TO.CENTER	4
RESET.GRAPH.EXTENT	2	WINDOW.SCALE	5		
SCALE.GRAPH	2	ZOOM.GRAPH.ADDLINKFN	3		

VARIABLE INDEX

IP.DECREASING.FONT.LIST	5	PRESS.DECREASING.FONT.LIST	5
-------------------------------	---	----------------------------------	---

RECORD INDEX

ZOOMGRAPH	1	ZOOMGRAPHNODE	1
-----------------	---	---------------------	---
