

File created: 20-May-2000 10:22:28 {DSK}<project>medley3.5>lispusers>PLOT.;3

changes to: (VARS PLOTCOMS)

previous date: 4-Nov-93 14:56:28 {DSK}<project>medley3.5>lispusers>PLOT.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
:: Copyright (c) 1985, 1986, 1987, 1988, 1993, 2000 by Xerox Corporation. All rights reserved.

(RPAQQ PLOTCOMS

[

::: PLOT manager fns

```
(FNS ADDPLOT OBJECT ADJUSTSCALE? ADJUSTVIEWPORT APPLY.AFTERFN.MACRO ASKFORLABEL ASKFORSCALE BOXREGION
CHOOSESCALE CHOOSETICS CLOSEPLOTWINDOW CLOSESTPLOT OBJECT COMPOUNDSUBTYPE COMPUTEBOTTOMMARGIN
COMPUTELEFTMARGIN COMPUTERIGHTMARGIN COMPUTETOPMARGIN COPYMENU CREATEPLOT CREATEPLOT FNS
CREATEPLOT OBJECT DEFAULTSCALEFN DEFAULTTICFN DEFAULTTICMETHOD DELETEPLOT OBJECT DESELECTPLOT OBJECT
DISTANCETOPLOT OBJECT DRAWBOTTOMMARGIN DRAWLEFTMARGIN DRAWMARGIN DRAWPLOT OBJECT DRAWPLOT
DRAWRIGHTMARGIN DRAWTOPMARGIN ERASEPLOT OBJECT EXTENDEDSCALEFN EXTENTOFPLOT OBJECT EXTENTOFPLOT
GETPLOTWINDOW GETTICLIST HIGHLIGHTPLOT OBJECT LABELPLOT OBJECT LOWLIGHTPLOT OBJECT MANUALRESCALE
MINSTREAMREGIONSIZE MOVEPLOT OBJECT OPENPLOTWINDOW PLOT.BUTTONEVENTFN PLOT.CLOSEFN PLOT.DEFAULTMENU
PLOT.FIXRIGHTMENU PLOT.HARDCOPYFN PLOT.ICONFN PLOT.LABELTOWORLD PLOT.REPAINTFN PLOT.RESET
PLOT.SETUP PLOT.SKETCH.CREATE PLOT.WHENSELECTEDFN PLOT.WORLDTOLABEL PLOT.ADDMENUITEMS PLOT.ADDPROP
PLOT.AXISINTERVAL PLOT.DELMENUITEMS PLOT.DELPROP PLOT.LABEL PLOT.MENU PLOT.MENUITEMS PLOT.OBJECTADDPROP
PLOT.OBJECTDELPROP PLOT.OBJECTLABEL PLOT.OBJECTPROP PLOT.OBJECTPROPMACRO PLOT.OBJECTSUBTYPE PLOT.OPERROR
PLOT.PROMPT PLOT.PROP PLOT.PROPMACRO PLOT.REMPROP PLOT.SCALEFN PLOT.TICFN PLOT.TICINFO PLOT.TICMETHOD
PLOT.TICS PRINTFN PRINTMENU REDRAWPLOTWINDOW RELABELSELECTEDPLOT OBJECT RESCALEPLOT SCALE
TOGGLABEL TOGGLEEXTENDEDAXES TOGGLEFIXEDMENU TOGGLETICS TRANSLATEPLOT OBJECT UNDELETEPLOT OBJECT
UNLABELPLOT OBJECT WHICHLABEL WHICHPLOT)
```

::: Fns to do our own number printing

```
(FNS PLOT.PRINTNUM PLOT.FNUM-STRING PLOT.ENUM-STRING CREATETICLISTS NORMALIZE-TICLIST)
(FNS DRAW-TICS-LEFT-RIGHT DRAW-TICS-TOP-BOTTOM DRAW-LABEL-LEFT-RIGHT DRAW-LABEL-TOP-BOTTOM)
(VARS PLOT.DEFAULTMIDDLEMENUITEMS PLOT.DEFAULTRIGHTMENUITEMS OBJECTOPSTABLE)
(RECORDS EXTENT MARGIN PLOT.PLOT FNS PLOT OBJECT AXISINFO AXISINTERVAL PLOTSCALE TICINFO)
(MACROS APPLY.AFTERFN PLOT OBJECTSUBTYPE? PLOT OBJECTPROP PLOT PROP)
(PROP ARG NAMES PLOT OBJECTPROP PLOT.DEFAULTMENU PLOT.FIXRIGHTMENU PLOT.LABEL PLOT.MENU PLOT.MENUITEMS
PLOT.PRETTYFNS PLOT.PROP PLOTSCALEFN PLOT.TICFN PLOT.TICS)
[INITVARS (SMALLPLOTFONT '(GACHA 8 MRR))
(LARGEPLTFONT '(GACHA 12 BRR))
```

::: PLOT I/O

```
(FNS COPYPLOT OBJECT COPYPLOT PLOT OBJECTPRINT PRINTPLOT OBJECT PRINTPLOT READFONT READMENU READPLOT OBJECT
READPLOT)
(FNS PRINT-VECTOR READ-VECTOR)
(FILEPKGCOMS PLOTS)
(ADDVARS (HPRINTMACROS (FONTDESCRIPTOR . PRINTFONT)
(MENU . PRINTMENU)
(PLOT . PRINTPLOT)
(PLOT OBJECT . PRINTPLOT OBJECT)
(ONED-ARRAY . PRINT-VECTOR)))
(ADDVARS (HPRINTREADFNS READPLOT READPLOT OBJECT READFONT READMENU READ-VECTOR))
(P (DEFPRINT 'PLOT OBJECT (FUNCTION PLOT OBJECTPRINT)))
```

::: Numeric fns

```
(FNS PLOT.EXP10 PLOT.LOG10 PLOT.FLOOR PLOT.CEILING SINEMAVE)
```

::: PLOT image object FNS

```
(FNS CREATEPLOTIMAGEOBJ CREATEPLOTBITMAPOBJ PLIO.BUTTONEVENTINFN PLIO.COPYFN PLIO.GETFN PLIO.PUTFN
PLIO.REINSERTOBJ PLOT.COPYBUTTONEVENTFN PLIO.DISPLAYFN PLIO.IMAGEBOXFN)
```

::: additional fns to allow plot im. objs. to work in Sketch

```
(FNS PLIO.EDITCLOSEFN IMAGE.OBJECT.CHANGED)
[INITVARS (PLOTIMAGEFNS (IMAGEFNSCREATE (FUNCTION PLIO.DISPLAYFN)
(FUNCTION PLIO.IMAGEBOXFN)
(FUNCTION PLIO.PUTFN)
(FUNCTION PLIO.GETFN)
(FUNCTION PLIO.COPYFN)
(FUNCTION PLIO.BUTTONEVENTINFN)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL))
```

```

(FUNCTION NIL)
(FUNCTION NIL)
(GLOBALVARS PLOTIMAGEFNS)

```

::: Initialize

```

(P (PLOT.SETUP OBJECTOPSTABLE)
 (PLOT.DEFAULTMENU 'MIDDLE PLOT.DEFAULTMIDDLEMENUITEMS)
 (PLOT.DEFAULTMENU 'RIGHT PLOT.DEFAULTRIGHTMENUITEMS))

```

::: Dependent files

```

(FILE TWODGRAPHICS PLOTOBJECTS)
(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
TWODGRAPHICS UNBOXEDOPS))
(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY (LOCALVARS . T))
(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
(ADDVARS (NLAMA)
(NLAML)
(LAMA PLOTTICS PLOTTICFN PLOTSCALEFN PLOTPROP PLOTOBJECTPROP PLOTMENUITEMS PLOTMENU
PLOTLABEL PLOT.FIXRIGHTMENU PLOT.DEFAULTMENU]))

```

::: PLOT manager fns

(DEFINEQ

**(ADDPLOTOBJECT**

```

[LAMBDA (OBJECT PLOT NODRAWFLG) ; Edited 5-May-87 18:11 by jop
 (PROG ((WHENADDEDFN (PLOTOBJECTPROP OBJECT 'WHENADDEDFN))
 REDRAWFLG NEWSCALES)
 [COND
 ((NOT (MEMB OBJECT (fetch PLOTOBJECTS of PLOT)))
 (replace PLOTOBJECTS of PLOT with (CONS OBJECT (fetch PLOTOBJECTS of PLOT))
 [COND
 ((ADJUSTSCALE? (EXTENTOFPLOTOBJECT OBJECT PLOT)
 PLOT)
 (SETQ REDRAWFLG T)))
 [COND
 (NULL NODRAWFLG)
 (COND
 (OR REDRAWFLG (NOT (OPENWP (fetch PLOTWINDOW of PLOT)
 (REDRAWPLOTWINDOW PLOT)))
 (T (DRAWPLOTOBJECT OBJECT (fetch PLOTWINDOWVIEWPORT of PLOT)
 PLOT])
 (APPLY.AFTERFN WHENADDEDFN OBJECT PLOT NODRAWFLG)
 (RETURN OBJECT]))

```

**(ADJUSTSCALE?**

```

[LAMBDA (EXTENT PLOT) ; Edited 5-May-87 18:12 by jop

```

::: Determines whether the plotting scale must be adjusted to included the extrema 'minx' , 'maxx' , etc. If so returns T. Side effects the ::: PLOTSCALE of PLOT

```

(LET* ((PLOTSCALE (fetch (PLOT PLOTSCALE) of PLOT))
 (XINTERVAL (fetch (PLOTSCALE XINTERVAL) of PLOTSCALE))
 (XAXISINFO (fetch (PLOTSCALE XAXISINFO) of PLOTSCALE))
 (XTICINFO (fetch (PLOTSCALE XTICINFO) of PLOTSCALE))
 (YINTERVAL (fetch (PLOTSCALE YINTERVAL) of PLOTSCALE))
 (YAXISINFO (fetch (PLOTSCALE YAXISINFO) of PLOTSCALE))
 (YTICINFO (fetch (PLOTSCALE YTICINFO) of PLOTSCALE))
 (MINX (fetch MINX of EXTENT))
 (MAXX (fetch MAXX of EXTENT))
 (MINY (fetch MINY of EXTENT))
 (MAXY (fetch MAXY of EXTENT))
 CHANGEDFLG)
 [COND
 ((OR (LESSP MINX (fetch (AXISINTERVAL MIN) of XINTERVAL))
 (GREATERP MAXX (fetch (AXISINTERVAL MAX) of XINTERVAL)))
 (SETQ CHANGEDFLG T)
 (LET [(NEWMIN (FMIN MINX (fetch (AXISINTERVAL MIN) of XINTERVAL)))
 (NEWMAX (FMAX MAXX (fetch (AXISINTERVAL MAX) of XINTERVAL))
 (SETQ XTICINFO (CHOOSETICS NEWMIN NEWMAX XAXISINFO PLOT))
 (SETQ XINTERVAL (CHOOSSCALE NEWMIN NEWMAX XAXISINFO XTICINFO PLOT))
 [COND
 ((OR (LESSP MINY (fetch (AXISINTERVAL MIN) of YINTERVAL))
 (GREATERP MAXY (fetch (AXISINTERVAL MAX) of YINTERVAL)))
 (SETQ CHANGEDFLG T)
 (LET [(NEWMIN (FMIN MINY (fetch (AXISINTERVAL MIN) of YINTERVAL)))
 (NEWMAX (FMAX MAXY (fetch (AXISINTERVAL MAX) of YINTERVAL))
 (SETQ YTICINFO (CHOOSETICS NEWMIN NEWMAX YAXISINFO PLOT))
 (SETQ YINTERVAL (CHOOSSCALE NEWMIN NEWMAX YAXISINFO YTICINFO PLOT))
 [COND
 (CHANGEDFLG (replace (PLOTSCALE XINTERVAL) of PLOTSCALE with XINTERVAL)
 (replace (PLOTSCALE XTICINFO) of PLOTSCALE with XTICINFO)

```

```
(replace (PLOTSCALE YINTERVAL) of PLOTSCALE with YINTERVAL)
(replace (PLOTSCALE YTCINFO) of PLOTSCALE with YTCINFO))
CHANGEDFLG])
```

(ADJUSTVIEWPORT

```
[LAMBDA (VIEWPORT STREAMREGION PLOT) ; Edited 5-May-87 18:12 by job
  (PROG ((PLOTSCALE (fetch PLOTSCALE of PLOT))
        (PARENTSTREAM (fetch PARENTSTREAM of VIEWPORT))
        BOTTOMMARGINSIZE LEFTMARGINSIZE RIGHTMARGINSIZE TOPMARGINSIZE)
    (SETQ BOTTOMMARGINSIZE (COMPUTEBOTTOMMARGIN PARENTSTREAM (fetch BOTTOMMARGIN of PLOT)
        PLOT))
    (SETQ LEFTMARGINSIZE (COMPUTELEFTMARGIN PARENTSTREAM (fetch LEFTMARGIN of PLOT)
        PLOT))
    (SETQ RIGHTMARGINSIZE (COMPUTERIGHTMARGIN PARENTSTREAM (fetch RIGHTMARGIN of PLOT)
        PLOT))
    (SETQ TOPMARGINSIZE (COMPUTETOPMARGIN PARENTSTREAM (fetch TOPMARGIN of PLOT)
        PLOT))
    [replace WORLDREGION of VIEWPORT with (CREATEREGION (fetch MIN of (fetch XINTERVAL of PLOTSCALE))
        (fetch MIN of (fetch YINTERVAL of PLOTSCALE))
        (fetch INTERVALLENGTH of (fetch XINTERVAL of PLOTSCALE))
        (fetch INTERVALLENGTH of (fetch YINTERVAL of PLOTSCALE))
    [replace STREAMSUBREGION of VIEWPORT with (CREATEREGION (PLUS (fetch LEFT of STREAMREGION)
        (CAR LEFTMARGINSIZE))
        (PLUS (fetch BOTTOM of STREAMREGION)
        (CDR BOTTOMMARGINSIZE))
        (IDIFFERENCE (fetch WIDTH of STREAMREGION)
        (IPLUS (CAR LEFTMARGINSIZE)
        (CAR RIGHTMARGINSIZE)))
        (IDIFFERENCE (fetch HEIGHT of STREAMREGION)
        (IPLUS (CDR BOTTOMMARGINSIZE)
        (CDR TOPMARGINSIZE))
    (COMPUTETRANSFORM VIEWPORT)
    (RETURN VIEWPORT])
```

(APPLY.AFTERFN.MACRO

```
[LAMBDA (ARGS) ; Edited 5-May-87 18:16 by job
  (PROG ((FNS (CAR ARGS))
        (ARGLST (CDR ARGS)))
    (RETURN `(if ,FNS
                then (if (AND (LISTP ,FNS)
                              (NEQ (CAR ,FNS)
                                    'LAMBDA))
                        then (for FN in ,FNS do (CL:FUNCALL FN ,@ARGLST))
                        else (CL:FUNCALL ,FNS ,@ARGLST])
```

(ASKFORLABEL

```
[LAMBDA (PLOT MARGINNAME) ; Edited 5-May-87 18:16 by job
  ;; Prompt for new label and make the required call to LABELPLOT
  [COND
    ((EQ MARGINNAME 'TITLE)
     (SETQ MARGINNAME 'TOP)]
  (PROG ((PLOTPROMPT (fetch PLOTPROMPTWINDOW of PLOT))
        (MARGIN (SELECTQ MARGINNAME
                        (BOTTOM (fetch BOTTOMMARGIN of PLOT))
                        (LEFT (fetch LEFTMARGIN of PLOT))
                        (TOP (fetch TOPMARGIN of PLOT))
                        (RIGHT (fetch RIGHTMARGIN of PLOT))
                        (HELP "ILLEGAL MARGIN NAME" MARGINNAME)))
        (PROMPT (SELECTQ MARGINNAME
                        (BOTTOM "BOTTOM MARGIN LABEL?")
                        (LEFT "LEFT MARGIN LABEL?")
                        (TOP "TITLE?")
                        (RIGHT "RIGHT MARGIN LABEL?")
                        (HELP "ILLEGAL MARGIN NAME" MARGINNAME)))
        LABEL NEWLABEL)
    (SETQ LABEL (fetch (MARGIN LABEL) of MARGIN))
    (TERPRI PLOTPROMPT)
    [SETQ NEWLABEL (PROMPTFORWARD PROMPT LABEL "Type a label" PLOTPROMPT NIL NIL
        (CHARCODE (EOL LF ESCAPE TAB)
    (COND
      ((AND (NEQ NEWLABEL LABEL)
            (NOT (STREQUAL NEWLABEL LABEL)))
       (PLOTLABEL PLOT MARGINNAME NEWLABEL])
```

(ASKFORSCALE

```
[LAMBDA (PLOT AXIS) ; Edited 5-May-87 18:16 by job
  (PROG ((PLOTPROMPT (fetch PLOTPROMPTWINDOW of PLOT))
        (LOWER (PLOT.WORLDTOLABEL (SELECTQ AXIS
                                     (X (fetch (PLOT XLOWER) of PLOT))
                                     (Y (fetch (PLOT YLOWER) of PLOT))
                                     (HELP "Illegal axis" AXIS))
        PLOT AXIS))
```

```

(UPPER (PLOT.WORLDTOLABEL (SELECTQ AXIS
                          (X (fetch (PLOT XUPPER) of PLOT))
                          (Y (fetch (PLOT YUPPER) of PLOT))
                          (HELP "Illegal axis" AXIS))
      PLOT AXIS)))
(TERPRI PLOTPROMPT)
(SETQ LOWER (PLOT.LABELTOWORLD [READ (OPENSTRINGSTREAM (PROMPTFORWARD (CONCAT AXIS " axis: From ")
                                                                    LOWER "Type a number" PLOTPROMPT NIL
                                                                    NIL (CHARCODE (EOL LF ESCAPE TAB)
                                                                    PLOT AXIS))
                                (SETQ UPPER (PLOT.LABELTOWORLD [READ (OPENSTRINGSTREAM (PROMPTFORWARD " to " UPPER "Type a number"
                                                                    PLOTPROMPT NIL NIL
                                                                    (CHARCODE (EOL LF ESCAPE TAB)
                                                                    PLOT AXIS))
                                (RETURN (CONS LOWER UPPER])

```

**(BOXREGION**

[LAMBDA (REGION STREAM) ; Edited 5-May-87 18:16 by jop

;; Draw a box around a region in STREAM

```

(PROG ((RLEFT (fetch LEFT of REGION))
      (RBOTTOM (fetch BOTTOM of REGION))
      (RRIGHT (fetch RIGHT of REGION))
      (RTOP (fetch TOP of REGION))
      (LINEWIDTH (DSPSCALE NIL STREAM)))
      (DRAWLINE RLEFT RBOTTOM RRIGHT RBOTTOM LINEWIDTH 'REPLACE STREAM)
      (DRAWLINE RRIGHT RBOTTOM RRIGHT RTOP LINEWIDTH 'REPLACE STREAM)
      (DRAWLINE RRIGHT RTOP RLEFT RTOP LINEWIDTH 'REPLACE STREAM)
      (DRAWLINE RLEFT RTOP RLEFT RBOTTOM LINEWIDTH 'REPLACE STREAM])

```

**(CHOOSESCALE**

[LAMBDA (MIN MAX AXISINFO TICINFO PLOT) ; Edited 5-May-87 18:25 by jop

```

(PROG ((SCALEFN (fetch (AXISINFO SCALEFN) of AXISINFO))
      NEWINTERVAL)
      [SETQ NEWINTERVAL (COND
                        (SCALEFN (CL:FUNCALL SCALEFN MIN MAX TICINFO PLOT))
                        (T (DEFAULTSCALEFN MIN MAX TICINFO))
                        (AND (NOT (type? AXISINTERVAL NEWINTERVAL))
                          (HELP "Not an AXISINTERVAL" NEWINTERVAL))
                        (RETURN NEWINTERVAL])

```

**(CHOOSETICS**

[LAMBDA (MIN MAX AXISINFO PLOT) ; Edited 5-May-87 18:25 by jop

```

(PROG ((TICFN (fetch (AXISINFO TICFN) of AXISINFO))
      NEWTICINFO)
      [SETQ NEWTICINFO (COND
                        (TICFN (CL:FUNCALL TICFN MIN MAX PLOT))
                        (T (DEFAULTTICFN MIN MAX))
                        (AND (NOT (type? TICINFO NEWTICINFO))
                          (HELP "Not a TICINFO" NEWTICINFO))
                        (RETURN NEWTICINFO])

```

**(CLOSEPLOTWINDOW**

[LAMBDA (PLOT) ; Edited 5-May-87 18:17 by jop

```

(LET [(PLOTWINDOW (fetch (PLOT PLOTWINDOW) of PLOT))
      (WHENCLOSEDFN (PLOTPROP PLOT 'WHENCLOSEDFN)]

```

;; Unfix the right menu

```

(PLOT.FIXRIGHTMENU PLOT NIL)

```

;; Cleanup Window Props

```

(COND
  ((WINDOWP PLOTWINDOW)
   (WINDOWPROP PLOTWINDOW 'PLOT NIL)
   (WINDOWDELPROP PLOTWINDOW 'REPAINTFN (FUNCTION PLOT.REPAINTFN))
   (WINDOWDELPROP PLOTWINDOW 'RESHAPEFN (FUNCTION PLOT.REPAINTFN))
   (WINDOWDELPROP PLOTWINDOW 'CLOSEFN (FUNCTION PLOT.CLOSEFN))
   (WINDOWPROP PLOTWINDOW 'BUTTONEVENTFN (FUNCTION TOTOPW))
   (WINDOWPROP PLOTWINDOW 'RIGHTBUTTONFN NIL)
   (WINDOWPROP PLOTWINDOW 'COPYBUTTONEVENTFN NIL)
   (WINDOWPROP PLOTWINDOW 'HARDCOPYFN NIL)
   (WINDOWPROP PLOTWINDOW 'ICONFN NIL)
   (CLOSEW PLOTWINDOW)
   (DETACHALLWINDOWS PLOTWINDOW)))

```

;; A user hook

```

(APPLY.AFTERFN WHENCLOSEDFN PLOT])

```

**(CLOSESTPLOT OBJECT**

[LAMBDA (PLOT STREAMPOSITION) ; Edited 5-May-87 18:17 by jop

```

(for OBJECT in (fetch PLOT OBJECTS of PLOT) smallest (DISTANCETOPLOT OBJECT STREAMPOSITION PLOT])

```

**(COMPOUNDSUBTYPE**

[LAMBDA (COMPOUNDOBJECT) ; Edited 5-May-87 18:18 by jop  
(**fetch** COMPOUNDTYPE **of** (**fetch** OBJECTDATA **of** COMPOUNDOBJECT])

**(COMPUTEBOTTOMMARGIN**

[LAMBDA (STREAM BOTTOMMARGIN PLOT) ; Edited 5-May-87 18:18 by jop

;; Returns a size cons pair (width . height) in streamcoordinates

(**DECLARE** (SPECVARS SMALLFONT LARGEFONT))  
(PROG ((SMALLFONT (FONTCREATE SMALLPLOTFONT NIL NIL NIL STREAM))  
(LARGEFONT (FONTCREATE LARGEFONT NIL NIL NIL STREAM))  
(TICS? (**fetch** (MARGIN TICS?) **of** BOTTOMMARGIN))  
(LABEL (**fetch** (MARGIN LABEL) **of** BOTTOMMARGIN))  
(WIDTH 0)  
SMALLASCENT LARGEHEIGHT HEIGHT)  
(SETQ SMALLASCENT (FONTPROP SMALLFONT 'ASCENT))  
(SETQ LARGEHEIGHT (FONTPROP LARGEFONT 'HEIGHT)) ; margin of at least one LARGEHEIGHT  
[SETQ HEIGHT (COND  
((OR TICS? LABEL)  
LARGEHEIGHT)  
(T (ITIMES 2 LARGEHEIGHT))  
[COND  
(TICS? (SETQ HEIGHT (IPLUS HEIGHT (ITIMES 3 SMALLASCENT))  
[COND  
(LABEL (SETQ HEIGHT (IPLUS HEIGHT (ITIMES 2 LARGEHEIGHT)))  
(SETQ WIDTH (STRINGWIDTH LABEL LARGEFONT))  
(RETURN (CONS WIDTH HEIGHT))

**(COMPUTELEFTMARGIN**

[LAMBDA (STREAM LEFTMARGIN PLOT) ; Edited 13-May-87 13:36 by jop

;; Returns a (width . height) pair

(**DECLARE** (SPECVARS PRXFLG SMALLPLOTFONT LARGEFONT))  
(PROG ((SMALLFONT (FONTCREATE SMALLPLOTFONT NIL NIL NIL STREAM))  
(LARGEFONT (FONTCREATE LARGEFONT NIL NIL NIL STREAM))  
(TICS? (**fetch** (MARGIN TICS?) **of** LEFTMARGIN))  
(TICLIST (**fetch** (MARGIN TICLIST) **of** LEFTMARGIN))  
(LABEL (**fetch** (MARGIN LABEL) **of** LEFTMARGIN))  
(HEIGHT 0)  
LARGEWIDTH SMALLWIDTH WIDTH)  
(SETQ SMALLWIDTH (STRINGWIDTH 'A SMALLFONT))  
(SETQ LARGEWIDTH (STRINGWIDTH 'A LARGEFONT))  
[SETQ WIDTH (COND  
((OR TICS? LABEL)  
LARGEWIDTH)  
(T (ITIMES 2 LARGEWIDTH))  
[COND  
(TICS? (SETQ WIDTH (IPLUS WIDTH (ITIMES 2 SMALLWIDTH))  
(**bind** TICWIDTH **for** TICPAIR **in** TICLIST **largest** (STRINGWIDTH (CDR TICPAIR)  
SMALLFONT)  
**finally** (RETURN \$\$EXTREME)]  
[COND  
(LABEL (SETQ WIDTH (IPLUS WIDTH (ITIMES 2 LARGEWIDTH)))  
(SETQ HEIGHT (ITIMES (NCHARS LABEL)  
(FONTPROP LARGEFONT 'HEIGHT))  
(RETURN (CONS WIDTH HEIGHT))

**(COMPUTERIGHTMARGIN**

[LAMBDA (STREAM RIGHTMARGIN PLOT) ; Edited 13-May-87 13:37 by jop

;; Returns a (width . height) pair

(**DECLARE** (SPECVARS PRXFLG SMALLFONT LARGEFONT))  
(PROG ((SMALLFONT (FONTCREATE SMALLPLOTFONT NIL NIL NIL STREAM))  
(LARGEFONT (FONTCREATE LARGEFONT NIL NIL NIL STREAM))  
(TICS? (**fetch** (MARGIN TICS?) **of** RIGHTMARGIN))  
(TICLIST (**fetch** (MARGIN TICLIST) **of** RIGHTMARGIN))  
(LABEL (**fetch** (MARGIN LABEL) **of** RIGHTMARGIN))  
(HEIGHT 0)  
SMALLWIDTH LARGEWIDTH WIDTH)  
(SETQ SMALLWIDTH (STRINGWIDTH 'A SMALLFONT))  
(SETQ LARGEWIDTH (STRINGWIDTH 'A LARGEFONT))  
[SETQ WIDTH (COND  
((OR TICS? LABEL)  
LARGEWIDTH)  
(T (ITIMES 2 LARGEWIDTH))  
[COND  
(TICS? (SETQ WIDTH (IPLUS WIDTH (ITIMES 2 SMALLWIDTH))  
(**for** TICPAIR **in** TICLIST **largest** (STRINGWIDTH (CDR TICPAIR)  
SMALLFONT)  
**finally** (RETURN \$\$EXTREME)]  
[COND  
(LABEL (SETQ WIDTH (IPLUS WIDTH (ITIMES 2 LARGEWIDTH)))  
(SETQ HEIGHT (ITIMES (NCHARS LABEL)

```
(FONTPROP LARGEFONT 'HEIGHT]
(RETURN (CONS WIDTH HEIGHT])
```

**(COMPUTETOPMARGIN**

```
[LAMBDA (STREAM TOPMARGIN PLOT) ; Edited 5-May-87 18:19 by jop
  (DECLARE (SPECVARS SMALLFONT LARGEFONT))
  (PROG ((SMALLFONT (FONTCREATE SMALLPLOTFONT NIL NIL NIL STREAM))
        (LARGEFONT (FONTCREATE LARGEPLTFONT NIL NIL NIL STREAM))
        (TICS? (fetch (MARGIN TICS?) of TOPMARGIN))
        (LABEL (fetch (MARGIN LABEL) of TOPMARGIN))
        (WIDTH 0)
        SMALLASCENT LARGEHEIGHT HEIGHT)
    (SETQ SMALLASCENT (FONTPROP SMALLFONT 'ASCENT))
    (SETQ LARGEHEIGHT (FONTPROP LARGEFONT 'HEIGHT)) ; margin of at least one LARGEHEIGHT
    [SETQ HEIGHT (COND
      ((OR TICS? LABEL)
       LARGEHEIGHT)
      (T (ITIMES 2 LARGEHEIGHT))
    ]
    [COND
      (TICS? (SETQ HEIGHT (IPLUS HEIGHT (ITIMES 3 SMALLASCENT))
    [COND
      (LABEL (SETQ HEIGHT (IPLUS HEIGHT (ITIMES 2 LARGEHEIGHT)))
      (SETQ WIDTH (IMAX WIDTH (STRINGWIDTH LABEL LARGEFONT))
    (RETURN (CONS WIDTH HEIGHT])
```

**(COPYMENU**

```
[LAMBDA (MENU NEWITEMS) ; Edited 5-May-87 18:19 by jop
  ;; Note that menu props are not copied
  (create MENU
    ITEMS _ (OR NEWITEMS (fetch ITEMS of MENU))
    WHENSELECTEDFN _ (fetch WHENSELECTEDFN of MENU)
    WHENHELDFN _ (fetch WHENHELDFN of MENU)
    WHENUNHELDFN _ (fetch WHENUNHELDFN of MENU)
    MENUPOSITION _ (fetch MENUPOSITION of MENU)
    MENUOFFSET _ (fetch MENUOFFSET of MENU)
    MENUFONT _ (fetch MENUFONT of MENU)
    MENUTITLEFONT _ (fetch MENUTITLEFONT of MENU)
    TITLE _ (fetch TITLE of MENU)
    CENTERFLG _ (fetch CENTERFLG of MENU)
    MENUBORDERSIZE _ (fetch MENUBORDERSIZE of MENU)
    MENUOUTLINE SIZE _ (fetch MENUOUTLINE SIZE of MENU)
    CHANGEOFFSETFLG _ (fetch CHANGEOFFSETFLG of MENU))
```

**(CREATEPLOT**

```
[LAMBDA (OPENFLG REGION TITLE BORDER) ; Edited 5-May-87 18:19 by jop
  ;; Creates a PLOT. If OPENFLG is T then the PLOT's associated window is opened. The other arguments are passed to CREATEW
  (PROG ((PLOT (create PLOT)))
    (replace (PLOT PLOTSCALE) of PLOT
      with (create PLOTSCALE
        XAXISINFO _ (create AXISINFO)
        XINTERVAL _ (create AXISINTERVAL
          MIN _ 0.0
          MAX _ 1.0)
        XTICINFO _ (create TICINFO
          TICMIN _ 0.0
          TICMAX _ 1.0
          TICINC _ 1.0
          NTICS _ 2)
        YAXISINFO _ (create AXISINFO)
        YINTERVAL _ (create AXISINTERVAL
          MIN _ 0.0
          MAX _ 1.0)
        YTICINFO _ (create TICINFO
          TICMIN _ 0.0
          TICMAX _ 1.0
          TICINC _ 1.0
          NTICS _ 2)))
    (PLOTMENU PLOT 'MIDDLE (PLOT.DEFAULTMENU 'MIDDLE))
    (PLOTMENU PLOT 'RIGHT (PLOT.DEFAULTMENU 'RIGHT)) ; Compute size of margins in stream coordinates
    (replace (PLOT BOTTOMMARGIN) of PLOT with (create MARGIN
      TICMETHOD _ 'DEFAULT))
    (replace (PLOT LEFTMARGIN) of PLOT with (create MARGIN
      TICMETHOD _ 'DEFAULT))
    (replace (PLOT TOPMARGIN) of PLOT with (create MARGIN
      TICMETHOD _ 'DEFAULT))
    (replace (PLOT RIGHTMARGIN) of PLOT with (create MARGIN
      TICMETHOD _ 'DEFAULT)) ; Cache display parameters until OPENPLOTWINDOW is called
    [COND
      ((OR REGION TITLE BORDER)
       (replace (PLOT PLOTWINDOW) of PLOT with (LIST REGION TITLE BORDER))
    [COND
```

```
(OPENFLG (OPENPLOTWINDOW PLOT)))
(RETURN PLOT])
```

(CREATEPLOTFNS

```
[LAMBDA (DRAWFN ERASEFN EXTENTFN DISTANCEFN HIGHLIGHTFN LOWLIGHTFN LABELFN MOVEFN COPYFN PUTFN GETFN
BORROWFROM) ; Edited 5-May-87 18:20 by jop
```

;; Create an instance of PLOTFNS, a vector of functions that implement generic plot object operations. A DRAWFN , ERASEFN , and a
;; EXTENTFN are required. If there is a DISTANCEFN then a HIGHLIGHTFN must also be supplied. Supplies defaults for some generic
;; operations. If BORROWFROM then it must be another PLOTFNS, in which case NIL functions are inherited from USING.

```
(DECLARE (SPECVARS DRAWFN ERASEFN EXTENTFN DISTANCEFN HIGHLIGHTFN LOWLIGHTFN LABELFN MOVEFN COPYFN PUTFN
GETFN))
[COND
(BORROWFROM [COND
((AND (NULL LOWLIGHTFN)
(NULL HIGHLIGHTFN))
(SETQ LOWLIGHTFN (fetch LOWLIGHTFN of BORROWFROM)
(for FN in '(DRAWFN ERASEFN EXTENTFN HIGHLIGHTFN LABELFN DISTANCEFN MOVEFN COPYFN PUTFN GETFN)
do (COND
((NULL (EVAL FN))
(SET FN (RECORDACCESS FN BORROWFROM)
(COND
((NOT (AND DRAWFN ERASEFN EXTENTFN))
(HELP "Attempt to create PLOTFNS without required FNS"))))
(COND
((AND DISTANCEFN (NOT HIGHLIGHTFN))
(HELP "DISTANCEFN without a HIGHLIGHTFN"))))
(create PLOTFNS
DRAWFN _ DRAWFN
ERASEFN _ ERASEFN
HIGHLIGHTFN _ (OR HIGHLIGHTFN (FUNCTION PLOTOPERROR))
LOWLIGHTFN _ (OR LOWLIGHTFN HIGHLIGHTFN (FUNCTION PLOTOPERROR))
MOVEFN _ (OR MOVEFN (FUNCTION PLOTOPERROR))
LABELFN _ (OR LABELFN (FUNCTION LABELGENERIC))
EXTENTFN _ EXTENTFN
DISTANCEFN _ [OR DISTANCEFN (FUNCTION (LAMBDA NIL MAX.SMALLP)
COPYFN _ (OR COPYFN (FUNCTION COPYGENERIC))
PUTFN _ (OR PUTFN (FUNCTION PUTGENERIC))
GETFN _ (OR GETFN (FUNCTION GETGENERIC))
```

(CREATEPLOTOBJECT

```
[LAMBDA (OBJECTFNS OBJECTSUBTYPE OBJECTLABEL OBJECTMENU OBJECTDATA) ; Edited 5-May-87 18:20 by jop
```

```
(COND
((NOT (AND OBJECTFNS OBJECTDATA))
(HELP "Attempt to create a PLOTOBJECT without a FNS vector or OBJECTDATA")))
(PROG ((PLOTOBJECT (create PLOTOBJECT
OBJECTFNS _ OBJECTFNS
OBJECTSUBTYPE _ OBJECTSUBTYPE
OBJECTLABEL _ OBJECTLABEL
OBJECTDATA _ OBJECTDATA))) ; PLOTOBJECTPROP coerces OBJECTMENU to a menu if it is
; an item list
(PLOTOBJECTPROP PLOTOBJECT 'OBJECTMENU OBJECTMENU)
(RETURN PLOTOBJECT])
```

(DEFAULTSCALEFN

```
[LAMBDA (MIN MAX TICINFO) ; Edited 5-May-87 18:20 by jop
```

```
(create AXISINTERVAL
MIN _ (fetch (TICINFO TICMIN) of TICINFO)
MAX _ (fetch (TICINFO TICMAX) of TICINFO])
```

(DEFAULTTICFN

```
[LAMBDA (MIN MAX TICS ROUND POWER) ; Edited 5-May-87 18:20 by jop
```

;; Computes an interval that includes (MIN,MAX) and can be exactly spanned by (NTICS-1) \*some increment. If TICS is NIL tries a few values and
;; chooses the one that yields the shortest interval.

```
(COND
[ (NULL TICS)
(SETQ TICS '(3 4 5 6 7 8))
((FIXP TICS)
(SETQ TICS (LIST TICS)))
(NLISTP TICS)
(HELP "Not a list of FIXP's" TICS))]
(bind (SHORTEST _ (SCALE MIN MAX (CAR TICS)
ROUND POWER))
CURRENT for NTICS in (CDR TICS) do (SETQ CURRENT (SCALE MIN MAX NTICS ROUND POWER))
(COND
((LESSP (fetch TICINTERVALLENGTH of CURRENT)
(fetch TICINTERVALLENGTH of SHORTEST))
(SETQ SHORTEST CURRENT)))
finally (RETURN SHORTEST])
```

**(DEFAULTTICMETHOD**

[LAMBDA (MARGIN PLOTSCALE PLOT) ; Edited 5-May-87 18:21 by jop

;; Return the default tic list based on the values of PLOTSCALE

```
(PROG ((TICINFO (SELECTQ MARGIN
  ((BOTTOM TOP)
    (fetch (PLOTSCALE XTICINFO) of PLOTSCALE))
  ((RIGHT LEFT)
    (fetch (PLOTSCALE YTICINFO) of PLOTSCALE))
  (HELP "MARGIN must be one of RIGHT, LEFT, TOP, BOTTOM" MARGIN)))
  TICINC)
  (SETQ TICINC (fetch (TICINFO TICINC) of TICINFO))
  (RETURN (COND
    ((LISTP TICINC)
      TICINC)
    ((NUMBERP TICINC)
      ; Be carefull that min and max tics correspond to min and max of
      ; interval
      (NCONC1 (for I from 1 to (SUB1 (fetch (TICINFO NTICS) of TICINFO)) as X
        from (fetch (TICINFO TICMIN) of TICINFO) by TICINC collect X)
        (fetch (TICINFO TICMAX) of TICINFO)))
    (T (HELP "Invalid TICINC" TICINC))
```

**(DELETEPLOT OBJECT**

[LAMBDA (OBJECT PLOT NODRAWFLG NOSAVEFLG) ; Edited 5-May-87 18:21 by jop

;; Delete object from display list of plot. If (NULL NODRAWFLG) then update the display (open it if necessary) if (NULL NOSAVEFLG) then intern the object on the save list.

```
(LET [(PLOT OBJECTS (fetch (PLOT PLOT OBJECTS) of PLOT))
  (PLOT WINDOW (fetch (PLOT PLOT WINDOW) of PLOT))
  (WHENDELETEDFN (PLOT OBJECTS PROP OBJECT 'WHENDELETEDFN))]
  (if (MEMB OBJECT PLOT OBJECTS)
    then (if (EQ OBJECT (fetch (PLOT SELECTED OBJECT) of PLOT))
      then (if (NULL NODRAWFLG)
        then (if (OPENWP PLOT WINDOW)
          then (LOWLIGHTPLOT OBJECT PLOT))
          (replace (PLOT SELECTED OBJECT) of PLOT with NIL))
        (replace (PLOT PLOT OBJECTS) of PLOT with (DREMOVE OBJECT PLOT OBJECTS))
        (if (NULL NOSAVEFLG)
          then (push (fetch (PLOT PLOTSAVE LIST) of PLOT)
            OBJECT))
        (if (NULL NODRAWFLG)
          then (if (NOT (OPENWP PLOT WINDOW))
            then (OPENPLOT WINDOW PLOT)
            else (ERASEPLOT OBJECT PLOT)))
        (APPLY .AFTERFN WHENDELETEDFN OBJECT PLOT NODRAWFLG NOSAVEFLG)
        OBJECT))
```

**(DESELECTPLOT OBJECT**

[LAMBDA (PLOT) ; Edited 5-May-87 18:21 by jop

```
(if (fetch (PLOT SELECTED OBJECT) of PLOT)
  then (LOWLIGHTPLOT (fetch (PLOT SELECTED OBJECT) of PLOT)
    PLOT)
  (replace (PLOT SELECTED OBJECT) of PLOT with NIL])
```

**(DISTANCETO PLOT OBJECT**

[LAMBDA (OBJECT STREAM POSITION PLOT) ; Edited 5-May-87 18:25 by jop

```
(CL:FUNCALL (fetch (PLOT FNS DISTANCE FN) of (fetch (PLOT OBJECT OBJECT FNS) of OBJECT))
  OBJECT STREAM POSITION PLOT])
```

**(DRAW BOTTOM MARGIN**

[LAMBDA (BOTTOM MARGIN STREAM VIEWPORT STREAM REGION PLOT) ; Edited 13-May-87 17:11 by jop

;; DRAW the BOTTOM MARGIN

```
(DECLARE (SPECVARS SMALLFONT LARGEFONT PRXFLG))
(PROG ((SMALLFONT (FONTCREATE SMALLPLOT FONT NIL NIL NIL STREAM))
  (LARGEFONT (FONTCREATE LARGE PLOT FONT NIL NIL NIL STREAM))
  (LABEL (fetch (MARGIN LABEL) of BOTTOM MARGIN))
  (XINTERVAL (fetch (PLOTSCALE XINTERVAL) of (fetch PLOTSCALE of PLOT)))
  SMALLPLOT FONT ASCENT BOTTOM)
  (SETQ SMALLPLOT FONT ASCENT (FONT PROP SMALLFONT 'ASCENT))
  (SETQ BOTTOM (fetch (REGION BOTTOM) of (fetch STREAM SUB REGION of VIEWPORT)))
  (if (fetch (MARGIN TICS?) of BOTTOM MARGIN)
    then ;; DRAW TICS and TIC labels if necessary
      (DRAW-TICS-TOP-BOTTOM (fetch (MARGIN TIC LIST) of BOTTOM MARGIN)
        (fetch MIN of XINTERVAL)
        (fetch MAX of XINTERVAL)
        (IPLUS SMALLPLOT FONT ASCENT BOTTOM)
        (IDIFFERENCE BOTTOM SMALLPLOT FONT ASCENT)
        (ITIMES 2 SMALLPLOT FONT ASCENT)
        SMALLFONT STREAM VIEWPORT T))
    (if LABEL
      then (DRAW-LABEL-TOP-BOTTOM LABEL LARGEFONT [PLUS (fetch (REGION BOTTOM) of STREAM REGION)
```



(IPLUS (FONTPROP STREAM 'DESCENT)
(FONTPROP LARGEFONT 'HEIGHT])

STREAMREGION STREAM])

(DRAWLEFTMARGIN

[LAMBDA (LEFTMARGIN STREAM VIEWPORT STREAMREGION PLOT) ; Edited 13-May-87 17:10 by jop

;; DRAW the BOTTOM MARGIN

(DECLARE (SPECVARS SMALLFONT LARGEFONT PRXFLG))

(PROG ((SMALLFONT (FONTCREATE SMALLPLOTFONT NIL NIL NIL STREAM))

(LARGEFONT (FONTCREATE LARGEPLTFONT NIL NIL NIL STREAM))

(LABEL (fetch (MARGIN LABEL) of LEFTMARGIN))

(YINTERVAL (fetch (PLOTSCALE YINTERVAL) of (fetch PLOTSCALE of PLOT)))

SMALLWIDTH LEFT)

(SETQ SMALLWIDTH (STRINGWIDTH 'A SMALLFONT))

(SETQ LEFT (fetch LEFT of (fetch STREAMSUBREGION of VIEWPORT)))

(if (fetch (MARGIN TICS?) of LEFTMARGIN)

then

;; DRAW TICS and TIC labels if necessary

(DRAW-TICS-LEFT-RIGHT (fetch (MARGIN TICLIST) of LEFTMARGIN)

(fetch MIN of YINTERVAL)

(fetch MAX of YINTERVAL)

(IPLUS SMALLWIDTH LEFT)

(IDIFFERENCE LEFT SMALLWIDTH)

SMALLWIDTH SMALLFONT STREAM VIEWPORT T))

(if LABEL

then

(DRAW-LABEL-LEFT-RIGHT LABEL LARGEFONT (PLUS (fetch (REGION LEFT) of STREAMREGION)

(STRINGWIDTH 'A LARGEFONT))

STREAMREGION STREAM])

(DRAWMARGIN

[LAMBDA (MARGIN STREAM STREAMVIEWPORT STREAMREGION PLOT) ; Edited 5-May-87 18:23 by jop

;; Draws the margin MARGIN (one of RIGHT LEFT BOTTOM or TOP)

(SELECTQ MARGIN

(RIGHT (DRAWRIGHTMARGIN (fetch RIGHTMARGIN of PLOT)

STREAM STREAMVIEWPORT STREAMREGION PLOT))

(LEFT (DRAWLEFTMARGIN (fetch LEFTMARGIN of PLOT)

STREAM STREAMVIEWPORT STREAMREGION PLOT))

(BOTTOM (DRAWBOTTOMMARGIN (fetch BOTTOMMARGIN of PLOT)

STREAM STREAMVIEWPORT STREAMREGION PLOT))

(TOP (DRAWTOPMARGIN (fetch TOPMARGIN of PLOT)

STREAM STREAMVIEWPORT STREAMREGION PLOT))

(HELP "MARGIN must be one of RIGHT, LEFT, BOTTOM, or TOP " MARGIN])

(DRAWPLOT OBJECT

[LAMBDA (OBJECT VIEWPORT PLOT) ; Edited 5-May-87 18:23 by jop

(PROG [(TEXTOBJECT (PLOT OBJECTPROP OBJECT 'LABEL))

(WHENDRAWNFN (PLOT OBJECTPROP OBJECT 'WHENDRAWNFN)

(CL:FUNCALL (fetch (PLOT FNS DRAWN) of (fetch (PLOT OBJECT OBJECTFNS) of OBJECT))

OBJECT VIEWPORT PLOT)

(COND

(TEXTOBJECT (DRAWPLOT OBJECT TEXT OBJECT VIEWPORT PLOT)))

(APPLY.AFTERFN WHENDRAWNFN OBJECT VIEWPORT PLOT])

(DRAWPLOT

[LAMBDA (PLOT CURRENTSTREAM STREAMVIEWPORT STREAMREGION) ; Edited 6-May-87 18:28 by jop

;; Draws a plot on CURRENTSTREAM. STREAMREGION is the region the PLOT will occupy. Does not blank the STREAMREGION before

;; drawing

(COND

((NOT (type? PLOT PLOT))

(HELP "Not a PLOT " PLOT)))

; Will not check, for the moment, that the streamregion is large

; enough

(BOXREGION (fetch STREAMSUBREGION of STREAMVIEWPORT)

CURRENTSTREAM)

(for MARGIN in '(BOTTOM LEFT TOP RIGHT) do (DRAWMARGIN MARGIN CURRENTSTREAM STREAMVIEWPORT STREAMREGION

PLOT))

(for OBJECT in (fetch PLOT OBJECTS of PLOT) do (DRAWPLOT OBJECT STREAMVIEWPORT PLOT])

(DRAWRIGHTMARGIN

[LAMBDA (RIGHTMARGIN STREAM VIEWPORT STREAMREGION PLOT) ; Edited 13-May-87 17:10 by jop

;; DRAW the RIGHT MARGIN

(DECLARE (SPECVARS SMALLFONT LARGEFONT PRXFLG))

(PROG ((SMALLFONT (FONTCREATE SMALLPLOTFONT NIL NIL NIL STREAM))

(LARGEFONT (FONTCREATE LARGEPLTFONT NIL NIL NIL STREAM))

(LABEL (fetch (MARGIN LABEL) of RIGHTMARGIN))

(YINTERVAL (fetch (PLOTSCALE YINTERVAL) of (fetch PLOTSCALE of PLOT)))

SMALLWIDTH RIGHT)

(SETQ SMALLWIDTH (STRINGWIDTH 'A SMALLFONT))

(SETQ RIGHT (fetch (REGION RIGHT) of (fetch STREAMSUBREGION of VIEWPORT)))

```

(if (fetch (MARGIN TICS?) of RIGHTMARGIN)
  then ;; DRAW TICS and TIC labels if necessary
    (DRAW-TICS-LEFT-RIGHT (fetch (MARGIN TICLIST) of RIGHTMARGIN)
      (fetch MIN of YINTERVAL)
      (fetch MAX of YINTERVAL)
      (IPLUS SMALLWIDTH RIGHT)
      (IDIFFERENCE RIGHT SMALLWIDTH)
      SMALLWIDTH SMALLFONT STREAM VIEWPORT))
  (if LABEL
    then (DRAW-LABEL-LEFT-RIGHT LABEL LARGEFONT (DIFFERENCE (fetch RIGHT of STREAMREGION)
      (ITIMES 2 (STRINGWIDTH 'A LARGEFONT)))
      STREAMREGION STREAM))

```

(DRAWTOPMARGIN

[LAMBDA (TOPMARGIN STREAM VIEWPORT STREAMREGION PLOT) ; Edited 13-May-87 17:11 by jop

```

;; DRAW the Top MARGIN
(DECLARE (SPECVARS SMALLFONT LARGEFONT PRXFLG))
(PROG ((SMALLFONT (FONTCREATE SMALLPLOTFONT NIL NIL NIL STREAM))
  (LARGEFONT (FONTCREATE LARGEPLTFONT NIL NIL NIL STREAM))
  (LABEL (fetch (MARGIN LABEL) of TOPMARGIN))
  (XINTERVAL (fetch (PLOTSCALE XINTERVAL) of (fetch PLOTSCALE of PLOT)))
  SMALLFONTASCENT TOP)
  (SETQ SMALLFONTASCENT (FONTPROP SMALLFONT 'ASCENT))
  (SETQ TOP (fetch TOP of (fetch STREAMSUBREGION of VIEWPORT)))
  (if (fetch (MARGIN TICS?) of TOPMARGIN)
    then ;; DRAW TICS and TIC labels if necessary
      (DRAW-TICS-TOP-BOTTOM (fetch (MARGIN TICLIST) of TOPMARGIN)
        (fetch MIN of XINTERVAL)
        (fetch MAX of XINTERVAL)
        (IPLUS SMALLFONTASCENT TOP)
        (IDIFFERENCE TOP SMALLFONTASCENT)
        SMALLFONTASCENT SMALLFONT STREAM VIEWPORT))
    (if LABEL
      then (DRAW-LABEL-TOP-BOTTOM LABEL LARGEFONT [IDIFFERENCE (fetch TOP of STREAMREGION)
        (IPLUS (FONTPROP LARGEFONT 'HEIGHT)
        (FONTPROP STREAM 'ASCENT]
        STREAMREGION STREAM))

```

(ERASEPLOT OBJECT

[LAMBDA (OBJECT PLOT) ; Edited 5-May-87 18:24 by jop

```

(PROG [(TEXTOBJECT (PLOT OBJECTPROP OBJECT 'LABEL))
  (WHENERASEDFN (PLOT OBJECTPROP OBJECT 'WHENERASEDFN))
  (CL:FUNCALL (fetch (PLOT FNS ERASEFN) of (fetch (PLOT OBJECT FNS) of OBJECT))
    OBJECT
    (fetch (PLOT PLOTWINDOWVIEWPORT) of PLOT)
    PLOT)
  (COND
    (TEXTOBJECT (ERASEPLOT OBJECT TEXTOBJECT PLOT))
  (APPLY.AFTERTFN WHENERASEDFN OBJECT PLOT])

```

(EXTENDEDSCALEFN

[LAMBDA (MIN MAX TICINFO) ; Edited 5-May-87 18:28 by jop

```

(PROG ((NEWMIN (fetch (TICINFO TICMIN) of TICINFO))
  (NEWMAX (fetch (TICINFO TICMAX) of TICINFO))
  (EPISILON 0.05)
  DELTA)
  (SETQ DELTA (FTIMES EPISILON (FDIFFERENCE NEWMAX NEWMIN)))
  (RETURN (create AXISINTERVAL
    MIN _ (FDIFFERENCE NEWMIN DELTA)
    MAX _ (FPLUS NEWMAX DELTA])

```

(EXTENTOFPLOT OBJECT

[LAMBDA (OBJECT PLOT) ; Edited 5-May-87 18:28 by jop

```

(CL:FUNCALL (fetch (PLOT FNS EXTENTFN) of (fetch (PLOT OBJECT FNS) of OBJECT))
  OBJECT PLOT])

```

(EXTENTOFPLOT

[LAMBDA (PLOT) ; Edited 5-May-87 18:28 by jop

```

(bind EXTENT (MINX _ MAX.FLOAT)
  (MAXX _ MIN.FLOAT)
  (MINY _ MAX.FLOAT)
  (MAXY _ MIN.FLOAT) for OBJECT in (fetch PLOT OBJECTS of PLOT)
  do (SETQ EXTENT (EXTENTOFPLOT OBJECT))
  [COND
    ((LESSP (fetch MINX of EXTENT)
      MINX)
      (SETQ MINX (fetch MINX of EXTENT))
    [COND
      ((GREATERP (fetch MAXX of EXTENT)

```

```

      MAXX)
    (SETQ MAXX (fetch MAXX of EXTENT])
[COND
  ((LESSP (fetch MINY of EXTENT)
    MINY)
    (SETQ MINY (fetch MINY of EXTENT])
[COND
  ((GREATERP (fetch MAXY of EXTENT)
    MAXY)
    (SETQ MAXY (fetch MAXY of EXTENT])
finally (RETURN (create EXTENT
      MINX _ MINX
      MAXX _ MAXX
      MINY _ MINY
      MAXY _ MAXY]))

```

**(GETPLOTWINDOW**

```

[LAMBDA (PLOT)
  (WINDOWP (fetch (PLOT PLOTWINDOW) of PLOT))

```

; Edited 5-May-87 18:29 by jop

**(GETTICLIST**

```

[LAMBDA (MARGINNAME PLOT)
  (LET* ((MARGIN (SELECTQ MARGINNAME
    (BOTTOM (fetch BOTTOMMARGIN of PLOT))
    (LEFT (fetch LEFTMARGIN of PLOT))
    (TOP (fetch TOPMARGIN of PLOT))
    (RIGHT (fetch RIGHTMARGIN of PLOT))
    (SHOULDNT)))
    (TICMETHOD (fetch TICMETHOD of MARGIN)))
    (COND
      ((EQ TICMETHOD 'DEFAULT)
        (DEFAULTTICMETHOD MARGINNAME (fetch PLOTSCALE of PLOT)
          PLOT))
      ((LITATOM TICMETHOD)
        (CL:FUNCALL TICMETHOD MARGINNAME (fetch PLOTSCALE of PLOT)
          PLOT))
      ((LISTP TICMETHOD)
        TICMETHOD)
      (T (HELP "Illegal ticmethod" TICMETHOD)))

```

; Edited 7-May-87 18:07 by jop

**(HIGHLIGHTPLOTOBJECT**

```

[LAMBDA (OBJECT PLOT)
  (PROG [(TEXTOBJECT (PLOTBJECTPROP OBJECT 'LABEL))
    (WHENHIGHLIGHTEDFN (PLOTBJECTPROP OBJECT 'WHENHIGHLIGHTEDFN]
    (CL:FUNCALL (fetch (PLOTFN HIGHLIGHTFN) of (fetch (PLOTBJECT OBJECTFN) of OBJECT))
      OBJECT
      (fetch (PLOT PLOTWINDOWVIEWPORT) of PLOT)
      PLOT)
    (COND
      (TEXTOBJECT (HIGHLIGHTPLOTBJECT TEXTOBJECT PLOT)))
    (APPLY.AFTERFN WHENHIGHLIGHTEDFN OBJECT PLOT])

```

; Edited 5-May-87 18:30 by jop

**(LABELPLOTBJECT**

```

[LAMBDA (OBJECT PLOT)
  (PROG [(WHENLABELEDFN (PLOTBJECTPROP OBJECT 'WHENLABELEDFN]
    (CL:FUNCALL (fetch (PLOTFN LABELFN) of (fetch (PLOTBJECT OBJECTFN) of OBJECT))
      OBJECT PLOT)
    (APPLY.AFTERFN WHENLABELEDFN OBJECT PLOT])

```

; Edited 5-May-87 18:30 by jop

**(LOWLIGHTPLOTBJECT**

```

[LAMBDA (OBJECT PLOT)
  (PROG [(TEXTOBJECT (PLOTBJECTPROP OBJECT 'LABEL))
    (WHENLOWLIGHTEDFN (PLOTBJECTPROP OBJECT 'WHENLOWLIGHTEDFN]
    (CL:FUNCALL (fetch (PLOTFN LOWLIGHTFN) of (fetch (PLOTBJECT OBJECTFN) of OBJECT))
      OBJECT
      (fetch (PLOT PLOTWINDOWVIEWPORT) of PLOT)
      PLOT)
    (COND
      (TEXTOBJECT (LOWLIGHTPLOTBJECT TEXTOBJECT PLOT)))
    (APPLY.AFTERFN WHENLOWLIGHTEDFN OBJECT PLOT])

```

; Edited 5-May-87 18:30 by jop

**(MANUALRESCALE**

```

[LAMBDA (PLOT AXIS)
  [COND
    ((NULL AXIS)
      (SETQ AXIS 'BOTH])
    (PROG ((PLOTSCALE (fetch PLOTSCALE of PLOT))
      (PLOTBJECTS (fetch PLOTBJECTS of PLOT))
      NEWSCALE)
    (COND
      ((OR (EQ AXIS 'BOTH)

```

; Edited 5-May-87 18:30 by jop

```

(EQ AXIS 'X)
(SETQ NEWSCALE (ASKFORSCALE PLOT 'X))
(COND
  ((GREATERP (CDR NEWSCALE)
    (CAR NEWSCALE))
   (LET ((NEWMIN (CAR NEWSCALE))
         (NEWMAX (CDR NEWSCALE))
         (AXISINFO (fetch (PLOTSCALE XAXISINFO) of PLOTSCALE)))
    (replace (PLOTSCALE XTICINFO) of PLOTSCALE with (CHOOSETICS NEWMIN NEWMAX AXISINFO PLOT))
    (replace (PLOTSCALE XINTERVAL) of PLOTSCALE
      with (create AXISINTERVAL
        MIN _ NEWMIN
        MAX _ NEWMAX])
   [COND
    ((OR (EQ AXIS 'BOTH)
         (EQ AXIS 'Y))
     (SETQ NEWSCALE (ASKFORSCALE PLOT 'Y))
     (COND
      ((GREATERP (CDR NEWSCALE)
        (CAR NEWSCALE))
       (LET ((NEWMIN (CAR NEWSCALE))
             (NEWMAX (CDR NEWSCALE))
             (AXISINFO (fetch (PLOTSCALE YAXISINFO) of PLOTSCALE)))
        (replace (PLOTSCALE YTICINFO) of PLOTSCALE with (CHOOSETICS NEWMIN NEWMAX AXISINFO PLOT))
        (replace (PLOTSCALE YINTERVAL) of PLOTSCALE
          with (create AXISINTERVAL
            MIN _ NEWMIN
            MAX _ NEWMAX])
       (REDRAWPLOTWINDOW PLOT]))

```

(MINSTREAMREGIONSIZE

```

[LAMBDA (STREAM PLOT) ; Edited 5-May-87 18:30 by jop
  ;; Compute the minimum acceptable size for a plot STREAMREGION. In the case of PLOTWINDOWS, corresponds to the min acceptable interior
  ;; size of the WINDOW. Returns a dotted pair (MINX . MINY) ; Sizes are (width . height) pairs
  (PROG ((BOTTOMMARGINSIZE (COMPUTELEFTMARGIN STREAM (fetch BOTTOMMARGIN of PLOT)
    PLOT))
        (LEFTMARGINSIZE (COMPUTELEFTMARGIN STREAM (fetch LEFTMARGIN of PLOT)
    PLOT))
        (RIGHTMARGINSIZE (COMPUTERIGHTMARGIN STREAM (fetch RIGHTMARGIN of PLOT)
    PLOT))
        (TOPMARGINSIZE (COMPUTETOPMARGIN STREAM (fetch TOPMARGIN of PLOT)
    PLOT))
        MINX MINY) ; The constant 100 is heuristic
    (SETQ MINX (IPLUS (CAR LEFTMARGINSIZE)
      (IMAX (CAR BOTTOMMARGINSIZE)
        (CAR TOPMARGINSIZE)
        100)
      (CAR RIGHTMARGINSIZE)))
    (SETQ MINY (IPLUS (CDR BOTTOMMARGINSIZE)
      (IMAX (CDR LEFTMARGINSIZE)
        (CDR RIGHTMARGINSIZE)
        100)
      (CDR TOPMARGINSIZE)))
    (RETURN (CONS MINX MINY))

```

(MOVEPLOT OBJECT

```

[LAMBDA (OBJECT DX DY PLOT) ; Edited 5-May-87 18:30 by jop
  (CL:FUNCALL (fetch (PLOTFNS MOVEFN) of (fetch (PLOT OBJECT OBJECTFNS) of OBJECT))
    OBJECT DX DY PLOT)

```

(OPENPLOTWINDOW

```

[LAMBDA (PLOT) ; Edited 19-May-87 10:17 by jop
  ;; Open window associated with PLOT. Creates circularities later broken by PLOT.CLOSEFN
  (COND
    ((NOT (type? PLOT PLOT))
     (HELP "Not a plot" PLOT))
    (PROG ((WINDOW (fetch (PLOT PLOTWINDOW) of PLOT))
          (PLOTWINDOW (fetch (PLOT PLOTWINDOW) of PLOT))
          (WHENOPENEDFN (PLOTPROP PLOT 'WHENOPENEDFN))
          MINSIZE WINDOWRESHAPEFLG PROMPTCREATEDFLG MINWINDOWEXTENT)
      (COND
        ((OPENWP WINDOW) ; No need to continue
         (RETURN WINDOW)))
      [COND
        ((NOT (WINDOWP WINDOW))
         (LET (REGION TITLE BORDER)
          [COND
            ((LISTP WINDOW)
             (SETQ REGION (CAR WINDOW))
             (SETQ TITLE (CADR WINDOW))
             (SETQ BORDER (CADDR WINDOW))
             (SETQ WINDOW (CREATEW (OR REGION (CREATEREGION 0 0 100 100))
              (OR TITLE "Plot Window"))

```



```

)
(replace (POSITION YCOORD) of POSITION with (SETQ NEWY (LASTMOUSEY
PLOTWINDOW))
)
[COND
  [(INSIDEP PLOTSUBREGION POSITION)
   (COND
     ((NOT (AND (EQ OLDX NEWX)
                (EQ OLDY NEWY)))
      (SETQ NEWSELECTEDOBJECT (CLOSESTPLOTBJECT PLOT POSITION
))
      (COND
        ((AND NEWSELECTEDOBJECT (NEQ NEWSELECTEDOBJECT
SELECTEDOBJECT))
         (COND
           (SELECTEDOBJECT (LOWLIGHTPLOTBJECT
SELECTEDOBJECT PLOT)))
          (HIGHLIGHTPLOTBJECT NEWSELECTEDOBJECT PLOT)
          (replace (PLOT SELECTEDOBJECT) of PLOT with
NEWSELECTEDOBJECT
)
          (SETQ SELECTEDOBJECT NEWSELECTEDOBJECT)
          ; Try to print a meaningful message in the
          ; PLOTPROMPTWINDOW
          (PLOTPROMPT (fetch (PLOTBJECT OBJECTLABEL)
of NEWSELECTEDOBJECT)
PLOT]
      (T (COND
          (SELECTEDOBJECT (LOWLIGHTPLOTBJECT SELECTEDOBJECT PLOT
)
          (SETQ SELECTEDOBJECT NIL)
          (replace (PLOT SELECTEDOBJECT) of PLOT with
SELECTEDOBJECT
)
          ]
          (SETQ OLDX NEWX)
          (SETQ OLDY NEWY]
  [(AND SELECTEDOBJECT (LASTMOUSESTATE MIDDLE))
   (LET ((MIDDLEMENU (fetch (PLOT MIDDLEMENU) of PLOT))
         (OBJECTMENU (fetch (PLOTBJECT OBJECTMENU) of SELECTEDOBJECT))
         MIDMENU)
       (SETQ MIDMENU (COND
         (OBJECTMENU [COND
           ((LITATOM OBJECTMENU)
            (SETQ OBJECTMENU (LISTGET (fetch (PLOT OTHERMENUS)
of PLOT)
OBJECTMENU]
           OBJECTMENU)
         (T MIDDLEMENU)))
       (COND
         (MIDMENU (PUTMENUPROP MIDMENU 'PLOT PLOT)
          (PUTMENUPROP MIDMENU 'MODE 'MIDDLE)
          (MENU MIDMENU)
          (PUTMENUPROP MIDMENU 'MODE NIL)
          (PUTMENUPROP MIDMENU 'PLOT NIL]
         ((LASTMOUSESTATE RIGHT)
          (LET [(RIGHTMENU (fetch (PLOT RIGHTMENU) of PLOT))
                (FIXEDRIGHTMENU? (PLOTPROP PLOT 'FIXEDRIGHTMENU?])
                (COND
                  ([OR FIXEDRIGHTMENU? (IGREATERP (fetch (POSITION YCOORD) of (CURSORPOSITION NIL PLOTWINDOW)
)
(WINDOWPROP PLOTWINDOW 'HEIGHT]
                  (DOWINDOWCOM PLOTWINDOW))
                  (RIGHTMENU (PUTMENUPROP RIGHTMENU 'PLOT PLOT)
                   (MENU RIGHTMENU)
                   (PUTMENUPROP RIGHTMENU 'PLOT NIL])

```

(PLOT.CLOSEFN

```

[LAMBDA (W)
  (CLOSEPLOTWINDOW (WINDOWPROP W 'PLOT))

```

; Edited 5-May-87 18:38 by jop

(PLOT.DEFAULTMENU

```

[LAMBDA ARGS

```

; Edited 5-May-87 18:38 by jop

:: If no third argument then simply return items list for given menu (middle or right), else replace the cached menu with the new list of items

```

(DECLARE (GLOBALVARS PLOT.DEFAULTMIDDLEMENU PLOT.DEFAULTRIGHTMENU))
(COND
  ((LESSP ARGS 1)
   (HELP "Must have at least one arg, MENUENAME"))
  (PROG ((MENUMAME (ARG ARGS 1))
         (NEWITEMS (AND (GREATERP ARGS 1)
                        (ARG ARGS 2)))
        MENU)
    (COND
      ((AND (GREATERP ARGS 1)
            (NOT (LISTP NEWITEMS)))

```

```

(HELP "Not a list" NEWITEMS)))
(SETQ MENU (SELECTQ MENUNAME
  (MIDDLE (AND (BOUNDP 'PLOT.DEFAULTMIDDLEMENU)
    PLOT.DEFAULTMIDDLEMENU))
  (RIGHT (AND (BOUNDP 'PLOT.DEFAULTRIGHTMENU)
    PLOT.DEFAULTRIGHTMENU))
  (SHOULDNT)))
[COND
  ((GREATERP ARGS 1)
  [SETQ MENU (AND NEWITEMS (COND
    (MENU (COPYMENU MENU NEWITEMS))
    (T (create MENU
      ITEMS _ NEWITEMS]
    (SELECTQ MENUNAME
      (MIDDLE (SETQ PLOT.DEFAULTMIDDLEMENU MENU))
      (RIGHT (SETQ PLOT.DEFAULTRIGHTMENU MENU))
      (SHOULDNT]
  (RETURN MENU])

```

(PLOT.FIXRIGHTMENU

; Edited 5-May-87 18:39 by job

```

[LAMBDA ARGS
  (COND
    ((ILESSP ARGS 1)
    (HELP "Must have at least one arg")))
  (LET* ((PLOT (ARG ARGS 1))
    [FIXEDFLG (COND
      ((IGREATERP ARGS 1)
      (ARG ARGS 2]
    (OLDVALUE (PLOTPROP PLOT 'FIXEDRIGHTMENU?))
    (PLOTWINDOW (fetch (PLOT PLOTWINDOW) of PLOT)))
  [COND
    ((IGREATERP ARGS 1)
    (LET [(FIXEDRIGHTMENU (WINDOWPROP PLOTWINDOW 'FIXEDRIGHTMENU)
      (PLOTPROP PLOT 'FIXEDRIGHTMENU? (NOT (NULL FIXEDFLG)))
      (COND
        [FIXEDFLG (COND
          ((AND (OPENWP PLOTWINDOW)
            (NULL FIXEDRIGHTMENU))
          (WINDOWPROP PLOTWINDOW 'FIXEDRIGHTMENU (ATTACHMENU (fetch (PLOT RIGHTMENU)
            of PLOT)
              PLOTWINDOW
              'RIGHT
              'TOP]
          (T (COND
            (FIXEDRIGHTMENU (CLOSEW FIXEDRIGHTMENU)
              (DETACHWINDOW FIXEDRIGHTMENU)
              (WINDOWPROP PLOTWINDOW 'FIXEDRIGHTMENU NIL]
          (OLDVALUE])

```

(PLOT.HARDCOPYFN

; Edited 13-May-87 12:27 by job

```

[LAMBDA (PLOTWINDOW PRINTERSTREAM)
  ;; Modified to allow hardcopy of plots on PRESS printers -- no landscape drawing
  ;; Modified to center plot on page
  (PROG ((WINDOWREGION (DSPCLIPPINGREGION NIL PLOTWINDOW))
    (PLOT (WINDOWPROP PLOTWINDOW 'PLOT))
    (VIEWPORT (CREATEVIEWPORT PRINTERSTREAM))
    PRINTERCLIPREGION STREAMREGION K)
  [if (EQ (IMAGESTREAMTYPE PRINTERSTREAM)
    'INTERPRESS)
    then (LET ((MICASPERINCH 2540))
      (if (GREATERP (fetch WIDTH of WINDOWREGION)
        (fetch HEIGHT of WINDOWREGION))
        then ; Print in landscape mode
          (ROTATE.IP PRINTERSTREAM 90)
          (CONCATT.IP PRINTERSTREAM)
          [TRANSLATE.IP PRINTERSTREAM 0 (FIX (MINUS (TIMES 8.5 MICASPERINCH]
            (CONCATT.IP PRINTERSTREAM) ; Make sure the clippingregion is rational
            (DSPCLIPPINGREGION (CREATEREGION (FIX (TIMES 0.5 MICASPERINCH))
              (FIX (TIMES 0.5 MICASPERINCH))
              (FIX (TIMES 10 MICASPERINCH))
              (FIX (TIMES 7.5 MICASPERINCH)))
            PRINTERSTREAM)
          else ; Make sure the clippingregion is rational
            (DSPCLIPPINGREGION (CREATEREGION (FIX (TIMES 0.5 MICASPERINCH))
              (FIX (TIMES 0.5 MICASPERINCH))
              (FIX (TIMES 7.5 MICASPERINCH))
              (FIX (TIMES 10 MICASPERINCH)))
            PRINTERSTREAM]
    (SETQ PRINTERCLIPREGION (DSPCLIPPINGREGION NIL PRINTERSTREAM))
    ; Reset the margins
    (DSPLEFTMARGIN (fetch (REGION LEFT) of PRINTERCLIPREGION)
      PRINTERSTREAM)
    (DSPBOTTOMMARGIN (fetch (REGION BOTTOM) of PRINTERCLIPREGION)
      PRINTERSTREAM)

```

```

(DSPRIGHTMARGIN (fetch (REGION RIGHT) of PRINTERCLIPREGION)
 PRINTERSTREAM)
(DSPTOPMARGIN (fetch (REGION TOP) of PRINTERCLIPREGION)
 PRINTERSTREAM) ; maintain the PLOTWINDOW's aspect ratio
[SETQ K (MIN (QUOTIENT (fetch (REGION WIDTH) of PRINTERCLIPREGION)
 (fetch (REGION WIDTH) of WINDOWREGION))
 (QUOTIENT (fetch (REGION HEIGHT) of PRINTERCLIPREGION)
 (fetch (REGION HEIGHT) of WINDOWREGION)))
 (SETQ STREAMREGION (LET [(SWIDTH (TIMES K (fetch (REGION WIDTH) of WINDOWREGION)))
 (SHEIGHT (TIMES K (fetch (REGION HEIGHT) of WINDOWREGION))
 ;; center plot on page
 (CREATEREGION (PLUS (fetch (REGION LEFT) of PRINTERCLIPREGION)
 (QUOTIENT (DIFFERENCE (fetch (REGION WIDTH) of
 PRINTERCLIPREGION
 SWIDTH)
 2))
 (PLUS (fetch BOTTOM of PRINTERCLIPREGION)
 (QUOTIENT (DIFFERENCE (fetch (REGION HEIGHT) of PRINTERCLIPREGION)
 SHEIGHT)
 2))
 SWIDTH SHEIGHT)))
 (CREATETICLISTS PLOT)
 (ADJUSTVIEWPORT VIEWPORT STREAMREGION PLOT)
 (DRAWPLOT PLOT PRINTERSTREAM VIEWPORT STREAMREGION)]

```

**(PLOT.ICONFN**

; Edited 5-May-87 18:40 by jop

```

[LAMBDA (PLOTWINDOW OLDICON)
 (PROG ((PLOT (WINDOWPROP PLOTWINDOW 'PLOT))
 (TITLEFONT (WINDOWTITLEFONT))
 ICONWIDTH ICONWHEIGHT SUBREGION ICONW VIEWPORT)
 (if (GREATERP (WINDOWPROP PLOTWINDOW 'WIDTH)
 (WINDOWPROP PLOTWINDOW 'HEIGHT))
 then (SETQ ICONWIDTH (WIDTHIFWINDOW 100))
 [SETQ ICONWHEIGHT (HEIGHTIFWINDOW (FIXR (TIMES 100 (FQUOTIENT (WINDOWPROP PLOTWINDOW
 'HEIGHT)
 (WINDOWPROP PLOTWINDOW 'WIDTH))
 (WINDOWPROP PLOTWINDOW 'HEIGHT))
 else [SETQ ICONWIDTH (WIDTHIFWINDOW (FIXR (TIMES 100 (FQUOTIENT (WINDOWPROP PLOTWINDOW 'WIDTH)
 (WINDOWPROP PLOTWINDOW 'HEIGHT))
 (SETQ ICONWHEIGHT (HEIGHTIFWINDOW 100)))
 (if OLDICON
 then (SHAPEW OLDICON (CREATEREGION (fetch LEFT of (WINDOWPROP OLDICON 'REGION))
 (fetch BOTTOM of (WINDOWPROP OLDICON 'REGION))
 ICONWIDTH ICONWHEIGHT))
 else (SETQ ICONW OLDICON)
 (SETQ ICONW (CREATEW (GETBOXREGION ICONWIDTH ICONWHEIGHT)))
 (DSPFONT TITLEFONT ICONW))
 (CLEARW ICONW)
 [SETQ SUBREGION (CREATEREGION [FIXR (TIMES 0.1 (WINDOWPROP ICONW 'WIDTH))
 [FIXR (TIMES 0.1 (WINDOWPROP ICONW 'HEIGHT))
 [FIXR (TIMES 0.8 (WINDOWPROP ICONW 'WIDTH))
 [FIXR (TIMES 0.8 (WINDOWPROP ICONW 'HEIGHT))
 [SETQ VIEWPORT (CREATEVIEWPORT (WINDOWPROP ICONW 'DSP)
 SUBREGION
 (fetch WORLDDREGION of (fetch PLOTWINDOWVIEWPORT of PLOT]
 (BOXREGION SUBREGION ICONW)
 [LET ((OBJECTS (fetch PLOT OBJECTS of PLOT))
 OBJECTS)
 (if (ILESSP (SETQ TOBJECTS (LENGTH OBJECTS))
 50)
 then ; few enough objects so that all of them may be drawn
 (for OBJECT in OBJECTS do (DRAWPLOT OBJECT VIEWPORT PLOT))
 else ; Sample the display list
 (bind (SAMPLERATE _ (FIXR (FQUOTIENT TOBJECTS 50))) for OBJECT in OBJECTS as I from 1
 when (IEQP 0 (IMOD I SAMPLERATE)) do (DRAWPLOT OBJECT VIEWPORT PLOT]
 (CENTERPRINTINREGION (OR (PLOT LABEL PLOT 'TOP)
 (if (NOT (STREQUAL (WINDOWPROP PLOTWINDOW 'TITLE)
 "Plot Window"))
 then (WINDOWPROP PLOTWINDOW 'TITLE)
 "Plot Icon")
 NIL ICONW)
 (RETURN ICONW])

```

**(PLOT.LABELTOWORLD**

; Edited 5-May-87 18:26 by jop

;; given label VALUE computes corresponding VALUE in world coords

```

(PROG [(FN (SELECTQ AXIS
 (X (PLOTPROP PLOT 'XWORLD FN))
 (Y (PLOTPROP PLOT 'YWORLD FN))
 (HELP "Illegal axis" AXIS]
 (RETURN (COND
 (FN (CL:FUNCALL FN VALUE PLOT AXIS))
 (T

```

; use identity transformation



VALUE])

(PLOT.REPAINTFN

[LAMBDA (WINDOW) ; Edited 5-May-87 18:40 by job
;; Redraws a PLOT WINDOW based on data stored on property list of WINDOW
(REDRAWPLOTWINDOW (WINDOWPROP WINDOW 'PLOT])

(PLOT.RESET

[LAMBDA (PLOT XSCALE YSCALE FLUSHMARGINS FLUSHPROPS NODRAWFLG) ; Edited 5-May-87 18:40 by job
;; Reset a PLOT for reuse. XSCALE must be an AXISINTERVAL, defaults to the current interval. Similarly for YSCALE. Non-NIL
;; FLUSHMARGINS means flush all labels, ticmethods, etc. Non-NIL FLUSHPROPS means flush all PLOTPROPS and cached menus
(if (NOT (type? PLOT PLOT))
then (HELP "NOT A PLOT" PLOT)) ; Flush display list
(replace (PLOT PLOTOBJECTS) of PLOT with NIL)
(replace (PLOT SELECTEDOBJECT) of PLOT with NIL)
(replace (PLOT PLOTSAVELIST) of PLOT with NIL)
(if FLUSHMARGINS
then (for MARGIN in '(BOTTOM LEFT TOP RIGHT) do (PLOTLABEL PLOT MARGIN NIL T)
(PLOTTICS PLOT MARGIN NIL T)
(PLOTTICMETHOD PLOT MARGIN NIL T)))
(if XSCALE
then (PLOTAXISINTERVAL PLOT 'X XSCALE T))
(if YSCALE
then (PLOTAXISINTERVAL PLOT 'Y YSCALE T)) ; Flush PLOT PROPS
(if FLUSHPROPS
then (replace (PLOT PLOTUSERDATA) of PLOT with NIL)
(replace (PLOT OTHERMENUS) of PLOT with NIL))
(if (NULL NODRAWFLG)
then (REDRAWPLOTWINDOW PLOT])

(PLOT.SETUP

[LAMBDA (OPSTABLE) ; Edited 7-May-87 18:28 by job
;; Assume opstable is a list of lists, one list for each PLOT object. The CAR of each sublist is the the name of the PLOT object, e.g. POINT. Then
;; follows pairs of method-names and function-names, e.g. (ADDFN ADDPOINTOBJECT)
[bind ASSOCLST for OBJECTLST in OPSTABLE
do (SET (PACK\* (CAR OBJECTLST)
'FNS)
(APPLY (FUNCTION CREATEPLOTFNS)
(first (SETQ ASSOCLST (CDR OBJECTLST)) for FNAME
in '(DRAWFN ERASEFN EXTENTFN DISTANCEFN HIGHLIGHTFN LOWLIGHTFN LABELFN MOVEFN COPYFN
PUTFN GETFN)
collect (CADR (ASSOC FNAME ASSOCLST])
(SETQ LARGEPLOTFONT (FONTCREATE LARGEPLOTFONT))
(SETQ SMALLPLOTFONT (FONTCREATE SMALLPLOTFONT])

(PLOT.SKETCH.CREATE

[LAMBDA (PLOT) ; Edited 5-May-87 18:41 by job
;; Creates a SKETCH STREAM and dumps the contents of PLOT into it
(if (NOT (type? PLOT PLOT))
then (HELP "Not a PLOT " PLOT))
(if (NOT (CL:FBOUND 'OPENSKECHSTREAM))
then (PLOT PROMPT "SKETCHSTREAM not loaded" PLOT)
else (PROG ([SKETCHSTREAM (OPENSKECHSTREAM "LAYOUT OF PLOT"
(if (fetch PLOTWINDOW of PLOT)
then (LET [(PLOTREGION (WINDOWPROP (fetch PLOTWINDOW of PLOT)
'REGION]
(LIST 'REGION (GETBOXREGION (fetch WIDTH of PLOTREGION)
(fetch HEIGHT of PLOTREGION]
SKETCHVIEWPORT)
(SETQ SKETCHVIEWPORT (CREATEVIEWPORT SKETCHSTREAM))
(ADJUSTVIEWPORT SKETCHVIEWPORT (DSPCLIPPINGREGION NIL SKETCHSTREAM)
PLOT)
(DRAWPLOT PLOT SKETCHSTREAM SKETCHVIEWPORT (DSPCLIPPINGREGION NIL SKETCHSTREAM])

(PLOT.WHENSELECTEDFN

[LAMBDA (ITEM MENU) ; Edited 5-May-87 18:42 by job
(LET\* ([PLOT (OR (GETMENUPROP MENU 'PLOT)
(WINDOWPROP (MAINWINDOW (WFROMMENU MENU))
'PLOT])
(MODE (GETMENUPROP MENU 'MODE))
(SELECTEDOBJECT (fetch (PLOT SELECTEDOBJECT) of PLOT))
(SELECTEDFN (CADR ITEM))
EXTRAARGS ARGSTOPASS)
[COND
((LISTP SELECTEDFN)
(SETQ EXTRAARGS (CDR SELECTEDFN))
(SETQ SELECTEDFN (CAR SELECTEDFN))
(SETQ ARGSTOPASS (for ARG in EXTRAARGS collect (EVAL ARG)))

```
(COND
  ((EQ MODE 'MIDDLE)
   (replace (PLOT SELECTEDOBJECT) of PLOT with NIL)
   (LOWLIGHTPLOT OBJECT SELECTEDOBJECT PLOT)
   (CL:APPLY SELECTEDFN SELECTEDOBJECT PLOT ARGSTOPASS))
  (T (CL:APPLY SELECTEDFN PLOT ARGSTOPASS)))
```

(PLOT.WORLDTOLABEL

[LAMBDA (VALUE PLOT AXIS) ; Edited 5-May-87 18:26 by jop

;; Given VALUE in world coords, computes corresponding label VALUE

```
(PROG [(FN (SELECTQ AXIS
            (X (PLOTPROP PLOT 'XLABELFN))
            (Y (PLOTPROP PLOT 'YLABELFN))
            (HELP "Illegal axis" AXIS]
      (RETURN (COND
                (FN (CL:FUNCALL FN VALUE PLOT AXIS))
                (T
                 VALUE]))
```

; use identity transformation

(PLOTADDMENUITEMS

[LAMBDA (PLOT MENUNAME ITEMSTOADD) ; Edited 5-May-87 18:42 by jop

;; Add ITEMSTOADD to end of menu MENUNAME item list

```
(PROG ((MENU (SELECTQ MENUNAME
              (MIDDLE (fetch MIDDLEMENU of PLOT))
              (RIGHT (fetch RIGHTMENU of PLOT))
              (LISTGET (fetch OTHERMENUS of PLOT)
                       MENUNAME)))
      (MENUITEMS (PLOTMENUITEMS PLOT MENUNAME)))
  (if ITEMSTOADD
    then (SETQ ITEMSTOADD (for ITEM in ITEMSTOADD unless (for ELEMENT in MENUITEMS
                                                           theirs (EQUAL (CAR ELEMENT)
                                                                    (CAR ITEM))))
          collect ITEM))
  (PLOTMENUITEMS PLOT MENUNAME (APPEND MENUITEMS ITEMSTOADD)))
(RETURN MENUITEMS))
```

(PLOTADDPROP

[LAMBDA (PLOT PROP ITEMTOADD FIRSTFLG) ; Edited 5-May-87 18:42 by jop

;; As in WINDOWADDPROP.

```
(PROG [(PROPVAL (MKLIST (PLOTPROP PLOT PROP))
      (if (NOT (MEMB ITEMTOADD PROPVAL))
          then (if FIRSTFLG
                 then (SETQ PROPVAL (CONS ITEMTOADD PROPVAL))
                 else (SETQ PROPVAL (APPEND PROPVAL (LIST ITEMTOADD)))
          (RETURN (PLOTPROP PLOT PROP PROPVAL]))
```

(PLOTAXISINTERVAL

[LAMBDA (PLOT AXIS INTERVAL NODRAWFLG) ; Edited 5-May-87 18:42 by jop

;; If INTERVAL is NIL returns the current INTERVAL for AXIS of PLOT. If INTERVAL is non-NIL it must be an INTERVAL, in which case the interval for axis AXIS of PLOT is set to INTERVAL

```
(PROG ((PLOTSCALE (fetch PLOTSCALE of PLOT))
      OLDVALUE)
  (SETQ OLDVALUE (SELECTQ AXIS
                        (X (fetch (PLOTSCALE XINTERVAL) of PLOTSCALE))
                        (Y (fetch (PLOTSCALE YINTERVAL) of PLOTSCALE))
                        (SHOULDNT)))
  (if (type? AXISINTERVAL INTERVAL)
    then (SELECTQ AXIS
              (X (replace (PLOTSCALE XINTERVAL) of PLOTSCALE with INTERVAL))
              (Y (replace (PLOTSCALE YINTERVAL) of PLOTSCALE with INTERVAL))
              (SHOULDNT))
      (if (NULL NODRAWFLG)
          then (REDRAWPLOTWINDOW PLOT)))
  (RETURN OLDVALUE]))
```

(PLOTDELMENUITEMS

[LAMBDA (PLOT MENUNAME ITEMSTODELETE) ; Edited 5-May-87 18:42 by jop

;; Delete ITEMSTODELETE from menu MENUNAME item list. RETURNS new item list if something deleted or else NIL. ITEMSTODELETE may be a list of lists or of atoms, in which case the atoms are compared to successive CARS of MENUNAME's item list

```
(SETQ ITEMSTODELETE (MKLIST ITEMSTODELETE))
(PROG ((MENU (SELECTQ MENUNAME
              (MIDDLE (fetch MIDDLEMENU of PLOT))
              (RIGHT (fetch RIGHTMENU of PLOT))
              (LISTGET (fetch OTHERMENUS of PLOT)
                       MENUNAME)))
      MENUITEMS SOMETHINGDELETED)
  (SETQ MENUITEMS (AND MENU (fetch ITEMS of MENU)))
```

```
[bind TARGET for ITEMTODELETE in ITEMSTODELETE
do (if (LITATOM ITEMTODELETE)
then (if [SETQ TARGET (for ITEM in MENUITEMS thereis (EQUAL ITEMTODELETE (CAR ITEM)]
then (SETQ SOMETHINGDELETED T)
(SETQ MENUITEMS (REMOVE TARGET MENUITEMS)))
elseif [AND (LISTP ITEMTODELETE)
(SETQ TARGET (CAR (MEMBER ITEMTODELETE MENUITEMS)]
then (SETQ SOMETHINGDELETED T)
(SETQ MENUITEMS (REMOVE TARGET MENUITEMS)]
(RETURN (if SOMETHINGDELETED
then (PLOTMENUITEMS PLOT MENUNAME MENUITEMS)
MENUITEMS]))
```

(PLOTDELPROP

```
[LAMBDA (PLOT PROP ITEMTODELETE) ; Edited 5-May-87 18:43 by jop
;; As in WINDOWDELPROP
(PROG ((PROPVAL (PLOTPROP PLOT PROP)))
(RETURN (if (EQ ITEMTODELETE PROPVAL)
then (PLOTPROP PLOT PROP NIL)
elseif (MEMB ITEMTODELETE PROPVAL)
then (PLOTPROP PLOT PROP (REMOVE ITEMTODELETE PROPVAL))
```

(PLOTLABEL

```
[LAMBDA ARGS ; Edited 25-Feb-88 13:49 by jop
;; IF NEWLABEL is not present then return current POSITION label of PLOT, else set the label to NEWLABEL and return the old value.
;; NODRAWFLG T suppresses redrawing. POSITION may be one of X, Y, TITLE
(COND
((LESSP ARGS 2)
(HELP "PLOTLABEL takes at least two args, plot and position")))
(PROG ((PLOT (ARG ARGS 1))
(POSITION (ARG ARGS 2))
(NEWLABEL (AND (GREATERP ARGS 2)
(ARG ARGS 3)))
(NODRAWFLG (AND (GREATERP ARGS 3)
(ARG ARGS 4)))
MARGIN OLDLABEL)
(SETQ MARGIN (SELECTQ POSITION
(BOTTOM (fetch BOTTOMMARGIN of PLOT))
(LEFT (fetch LEFTMARGIN of PLOT))
(TOP (fetch TOPMARGIN of PLOT))
(RIGHT (fetch RIGHTMARGIN of PLOT))
(HELP "Illegal margin" POSITION)))
(SETQ OLDLABEL (fetch (MARGIN LABEL) of MARGIN))
[COND
((GREATERP ARGS 2)
(replace (MARGIN LABEL) of MARGIN with (AND NEWLABEL (MKSTRING NEWLABEL)))
(COND
((NULL NODRAWFLG)
(REDRAWPLOTWINDOW PLOT]
(RETURN OLDLABEL]))
```

(PLOTMENU

```
[LAMBDA ARGS (* jop%: "12-Dec-85 10:31")
```

(\* If no third argument then simply return items list for given menu (middle or right)%, else replace the cached menu with the new list of items. If the NEWMENU's whenselectedfn is NIL it is replaced with PLOT.WHENSELECTEDFN)

```
(COND
((ILESSP ARGS 2)
(HELP "Must have at least two args, PLOT and MENUNAME")))
(PROG ((PLOT (ARG ARGS 1))
(MENUNAME (ARG ARGS 2))
(NEWMENU (AND (IGREATERP ARGS 2)
(ARG ARGS 3)))
PLOTWINDOW OLDVALUE)
(SETQ PLOTWINDOW (fetch (PLOT PLOTWINDOW) of PLOT))
(SETQ OLDVALUE (SELECTQ MENUNAME
(MIDDLE (fetch MIDDLEMENU of PLOT))
(RIGHT (fetch RIGHTMENU of PLOT))
(LISTGET (fetch OTHERMENUS of PLOT)
MENUNAME)))
[COND
((NOT (OR (NULL NEWMENU)
(type? MENU NEWMENU)))
(HELP "Not a menu" NEWMENU))
((AND NEWMENU (NULL (fetch WHENSELECTEDFN of NEWMENU)))
(replace (MENU WHENSELECTEDFN) of NEWMENU with (FUNCTION PLOT.WHENSELECTEDFN]
[COND
((IGREATERP ARGS 2)
[SELECTQ MENUNAME
(MIDDLE (replace MIDDLEMENU of PLOT with NEWMENU))
```

```

(RIGHT (replace RIGHTMENU of PLOT with NEWMENU))
(COND
  ((NULL (fetch OTHERMENUS of PLOT))
   (replace OTHERMENUS of PLOT with (LIST MENUNAME NEWMENU))
   NEWMENU)
  (T (LISTPUT (fetch OTHERMENUS of PLOT)
              MENUNAME NEWMENU]
(COND
  ((AND (OPENWP PLOTWINDOW)
        (EQ MENUNAME 'RIGHT)
        (PLOTPROP PLOT 'FIXEDRIGHTMENU?))
   (* Update the fixed menu)
   (PLOT.FIXRIGHTMENU PLOT NIL)
   (PLOT.FIXRIGHTMENU PLOT T]
(RETURN OLDVALUE])

```

**(PLOTMENUITEMS**

[LAMBDA ARGS (\* jop%: "11-Dec-85 14:39")

(\* If no third argument then simply return items list for given menu (middle or right)%, else replace the cached menu with the new list of items)

```

(if (LESSP ARGS 2)
  then (HELP "Must have at least two args, PLOT and MENUNAME"))
(PROG ((PLOT (ARG ARGS 1))
      (MENUNAME (ARG ARGS 2))
      (NEWITEMS (AND (GREATERP ARGS 2)
                    (ARG ARGS 3)))
      MENU)
  (if (AND (GREATERP ARGS 2)
          (NOT (LISTP NEWITEMS)))
    then (HELP "Not a list" NEWITEMS))
  (SETQ MENU (SELECTQ MENUNAME
                    (MIDDLE (fetch MIDDLEMENU of PLOT))
                    (RIGHT (fetch RIGHTMENU of PLOT))
                    (LISTGET (fetch OTHERMENUS of PLOT)
                             MENUNAME)))
  (if (GREATERP ARGS 2)
    then [SETQ MENU (AND NEWITEMS (if MENU
                                     then (COPYMENU MENU NEWITEMS)
                                     else (create MENU
                                                ITEMS _ NEWITEMS]
    (PLOTMENU PLOT MENUNAME MENU))
  (RETURN (if (LESSP ARGS 3)
              then (if MENU
                    then (fetch ITEMS of MENU))
              else NEWITEMS])

```

**(PLOTOBJECTADDPROP**

[LAMBDA (OBJECT PROP ITEMTOADD FIRSTFLG) (\* jop%: "20-Jan-86 16:03")

(\* As in WINDOWADDPROP.)

```

(PROG [(PROPVAL (MKLIST (PLOTOBJECTPROP OBJECT PROP]
  [if (NOT (MEMB ITEMTOADD PROPVAL))
    then (if FIRSTFLG
            then (SETQ PROPVAL (CONS ITEMTOADD PROPVAL))
            else (SETQ PROPVAL (APPEND PROPVAL (LIST ITEMTOADD]
  (RETURN (PLOTOBJECTPROP OBJECT PROP PROPVAL])

```

**(PLOTOBJECTDELPROP**

[LAMBDA (OBJECT PROP ITEMTODELETE) (\* jop%: "20-Jan-86 16:03")

(\* As in WINDOWDELPROP)

```

(PROG ((PROPVAL (PLOTOBJECTPROP OBJECT PROP)))
  (RETURN (if (EQ ITEMTODELETE PROPVAL)
              then (PLOTOBJECTPROP OBJECT PROP NIL)
              elseif (MEMB ITEMTODELETE PROPVAL)
              then (PLOTOBJECTPROP OBJECT PROP (REMOVE ITEMTODELETE PROPVAL])

```

**(PLOTOBJECTLABEL**

[LAMBDA (OBJECT LABEL PLOT NODRAWFLG) (\* edited%: "27-Mar-86 21:29")

(\* IF LABEL is NIL then return current label of OBJECT, else set the label to LABEL and return the old value. NODRAWFLG T suppresses drawing)

```

(if (NOT (type? PLOTOBJECT OBJECT))
  then (HELP "NOT A PLOTOBJECT" OBJECT))
(PROG ((OLDLABEL (fetch (PLOTOBJECT OBJECTLABEL) of OBJECT)))
  (if LABEL
    then (if (AND (NULL NODRAWFLG)
                  (PLOTOBJECTPROP OBJECT 'LABEL)
                  PLOT)

```

```

      then (UNLABELPLOT OBJECT PLOT))
    (replace (PLOT OBJECT OBJECTLABEL) of OBJECT with LABEL)
    (if (AND PLOT (NULL NODRAWFLG))
      then (LABELPLOT OBJECT PLOT)))
  (RETURN OLDLABEL])

```

**(PLOTOBJECTPROP**

[LAMBDA ARGS ; Edited 5-May-87 18:43 by jop  
 ;; As in WINDOWPROP. Operates on field OBJECTUSERDATA of PLOT OBJECT. If PROP is (QUOTE MENU) then accesses the object menu

```

(COND
  ((LESSP ARGS 2)
    (HELP "OBJECTPROP takes at least two arguments, plotobject and prop")))
  (PROG ((PLOT OBJECT (ARG ARGS 1))
    (PROPNAME (ARG ARGS 2))
    (NEWVALUE (AND (GREATERP ARGS 2)
      (ARG ARGS 3)))
    (FIELDNAMES '(OBJECTMENU OBJECTLABEL OBJECTDATA))
    OLDVALUE OBJECTUSERDATA)
    (SETQ OBJECTUSERDATA (fetch (PLOT OBJECT OBJECTUSERDATA) of PLOT OBJECT))
    [SETQ OLDVALUE (COND
      ((MEMB PROPNAME FIELDNAMES)
        (SELECTQ PROPNAME
          (OBJECTMENU (fetch (PLOT OBJECT OBJECTMENU) of PLOT OBJECT))
          (OBJECTLABEL (fetch (PLOT OBJECT OBJECTLABEL) of PLOT OBJECT))
          (OBJECTDATA (fetch (PLOT OBJECT OBJECTDATA) of PLOT OBJECT))
          (SHOULDNT)))
        (T (LISTGET OBJECTUSERDATA PROPNAME))
      ]
      ((GREATERP ARGS 2)
        (COND
          ((MEMB PROPNAME FIELDNAMES)
            (SELECTQ PROPNAME
              (OBJECTMENU (replace (PLOT OBJECT OBJECTMENU) of PLOT OBJECT
                with (OR [COND
                  ((LISTP NEWVALUE)
                    (COND
                      ((type? MENU OLDVALUE)
                        (LET ((NEWMENU (COPYMENU OLDVALUE NEWVALUE)))
                          [COND
                            ((NULL (fetch WHENSELECTEDFN of NEWMENU))
                              (replace WHENSELECTEDFN of NEWMENU
                                with (FUNCTION PLOT.WHENSELECTEDFN)
                                NEWMENU))
                            (T (create MENU
                              ITEMS _ NEWVALUE
                              WHENSELECTEDFN _ (FUNCTION PLOT.WHENSELECTEDFN)
                              NEWVALUE)))
                          ]
                        (OBJECTLABEL (replace (PLOT OBJECT OBJECTLABEL) of PLOT OBJECT with NEWVALUE))
                        (OBJECTDATA (replace (PLOT OBJECT OBJECTDATA) of PLOT OBJECT with NEWVALUE))
                        (SHOULDNT)))
                    (T (COND
                      ((NULL OBJECTUSERDATA)
                        (replace (PLOT OBJECT OBJECTUSERDATA) of PLOT OBJECT with (LIST PROPNAME NEWVALUE)))
                      (T (LISTPUT OBJECTUSERDATA PROPNAME NEWVALUE)
                        (RETURN OLDVALUE))
                    )
                ]
          )
        ]
      ]
    )
  )

```

**(PLOTOBJECTPROPMACRO**

[LAMBDA (ARGS) ; Edited 5-May-87 18:44 by jop

```

  (LET [(B PLOT OBJECT (CAR ARGS))
    (B PROPNAME (CADR ARGS))
    (FIELDNAMES '(OBJECTMENU OBJECTLABEL OBJECTDATA))
    (COND
      ((OR (NOT (EQLENGTH ARGS 2))
        (NEQ (CAR B PROPNAME)
          'QUOTE)
        (MEMB (CADR B PROPNAME)
          FIELDNAMES))
        'IGNOREMACRO)
      (T `(LISTGET (fetch (PLOT OBJECT OBJECTUSERDATA) of ,B PLOT OBJECT)
        ,B PROPNAME]))
    )
  )

```

**(PLOTOBJECTSUBTYPE**

[LAMBDA (PLOT OBJECT) (\* jop%: "20-Jan-86 16:21")  
 (fetch (PLOT OBJECT OBJECTSUBTYPE) of PLOT OBJECT])

**(PLOTOPERROR**

[LAMBDA NIL (\* edited%: "19-May-85 13:48")  
 (HELP "ATTEMPT To APPLY a generic PLOT operation to a deficient PLOT OBJECT")]

**(PLOTPROMPT**

[LAMBDA (TEXT PLOT) (\* jop%: " 3-Mar-85 15:42")

```
(PROG ((PLOT PROMPT WINDOW (fetch PLOT PROMPT WINDOW of PLOT)))
      (printout PLOT PROMPT WINDOW T TEXT])
```

**(PLOT PROP)**

[LAMBDA ARGS

; Edited 5-May-87 18:45 by jop

;; As in WINDOW PROP. See also PLOT PROP MACRO

```
(COND
  ((LESSP ARGS 2)
   (HELP "PLOT PROP TAKES AT LEAST TWO ARGUMENTS, PLOT and PROPNAME"))
  (PROG ((PLOT (ARG ARGS 1))
         (PROPNAME (ARG ARGS 2))
         (NEWVALUE (AND (GREATERP ARGS 2)
                        (ARG ARGS 3)))
         (FIELDS ' (XLOWER XUPPER YLOWER YUPPER MIDDLEMENU RIGHTMENU OTHERMENUS LEFTMARGIN RIGHTMARGIN
                    TOPMARGIN BOTTOMMARGIN PLOT WINDOW PLOT WINDOW VIEWPORT PLOT PROMPT WINDOW PLOT OBJECTS
                    PLOT SCALE SELECTED OBJECT WINDOW INFO MARGIN INFO MENU INFO PLOT USER DATA PLOT SAVE LIST))
         (OLDVALUE USERDATA)
```

;; FIELDS is given as an explicit LIST for efficiency reasons -- RECORD FIELD NAMES, although more robust, takes too long

```
(SETQ USERDATA (fetch (PLOT PLOT USER DATA) of PLOT))
[SETQ OLDVALUE (COND
  ((MEMB PROPNAME FIELDS)
   (RECORD ACCESS PROPNAME PLOT))
  (T (LISTGET USERDATA PROPNAME))
)]
[COND
  ((GREATERP ARGS 2)
   (COND
    ((MEMB PROPNAME FIELDS)
     (RECORD ACCESS PROPNAME PLOT NIL 'REPLACE NEWVALUE))
    (T (COND
        ((NULL USERDATA)
         (replace (PLOT PLOT USER DATA) of PLOT with (LIST PROPNAME NEWVALUE)))
        (T (LISTPUT USERDATA PROPNAME NEWVALUE))
       ))
     )
   )
  (RETURN OLDVALUE)]
```

**(PLOT PROP MACRO)**

[LAMBDA (ARGS)

; Edited 5-May-87 18:47 by jop

```
(LET [(B PLOT (CAR ARGS))
      (B PROPNAME (CADR ARGS))
      (B VALUE (CADDR ARGS))
      (FIELD NAMES ' (XLOWER XUPPER YLOWER YUPPER MIDDLEMENU RIGHTMENU OTHERMENUS LEFTMARGIN RIGHTMARGIN
                     TOPMARGIN BOTTOMMARGIN PLOT WINDOW PLOT WINDOW VIEWPORT PLOT PROMPT WINDOW PLOT OBJECTS
                     PLOT SCALE SELECTED OBJECT WINDOW INFO MARGIN INFO MENU INFO PLOT USER DATA PLOT SAVE LIST))]
  (if (NEQ (CAR B PROPNAME)
           'QUOTE)
      then 'IGNORE MACRO
      else (if (MEMB (CADR B PROPNAME)
                    FIELD NAMES)
              then [if (EQLLENGTH ARGS 3)
                      then `(PROG1 (fetch (PLOT , (CADR B PROPNAME)) of ,B PLOT)
                                   (replace (PLOT , (CADR B PROPNAME)) of ,B PLOT with ,B VALUE))
                      else `(fetch (PLOT , (CADR B PROPNAME)) of ,B PLOT]
              else (if (NOT (EQLLENGTH ARGS 2))
                      then 'IGNORE MACRO
                      else `(LISTGET (fetch (PLOT PLOT USER DATA) of ,B PLOT)
                                     ,B PROPNAME)]
```

**(PLOT REM PROP)**

[LAMBDA (PLOT PROPNAME)

; Edited 5-May-87 18:47 by jop

;; Destructively removes PROPNAME from proplist of PLOT

```
(if (NOT (type? PLOT PLOT))
  then (HELP "Not a plot" PLOT))
(PROG ((FIELDS (RECORD FIELD NAMES 'PLOT))
      (USERDATA (fetch (PLOT PLOT USER DATA) of PLOT))
      (LSTPTR OLDVALUE)
      (SETQ OLDVALUE (if (MEMB PROPNAME FIELDS)
                        then (RECORD ACCESS PROPNAME PLOT)
                        else (LISTGET USERDATA PROPNAME))))
  [if (MEMB PROPNAME FIELDS)
      then (RECORD ACCESS PROPNAME PLOT NIL 'REPLACE NIL)
      else (if (SETQ LSTPTR (MEMB PROPNAME USERDATA))
              then
                (if (EQ LSTPTR USERDATA)
                    then (replace (PLOT PLOT USER DATA) of PLOT with (CDDR USERDATA))
                    else (RPLACD (NLEFT USERDATA 1 LSTPTR)
                                (CDDR LSTPTR)))
              )
      )
  (RETURN OLDVALUE)]
```

; Splice out the offending links

**(PLOT SCALE FN)**

[LAMBDA ARGS

; Edited 5-May-87 18:47 by jop

(\* \*)

```
(COND
  ((ILESSP ARGS 2)
   (HELP "Must have at least two args")))
(PROG ((PLOT (ARG ARGS 1))
      (AXIS (ARG ARGS 2))
      AXISINFO OLDVALUE)
  (SETQ AXISINFO (SELECTQ AXIS
    (X (fetch (PLOTSCALE XAXISINFO) of (fetch PLOTSCALE of PLOT)))
    (Y (fetch (PLOTSCALE YAXISINFO) of (fetch PLOTSCALE of PLOT)))
    (SHOULDNT)))
  (SETQ OLDVALUE (fetch (AXISINFO SCALEFN) of AXISINFO))
  [COND
    ((IGREATERP ARGS 2)
     (LET [(NEWVALUE (ARG ARGS 3))
          (NODRAWFLG (AND (IGREATERP ARGS 3)
                        (ARG ARGS 4))
                (replace (AXISINFO SCALEFN) of AXISINFO with NEWVALUE)
                (RESCALEPLOT PLOT AXIS NODRAWFLG]
          (RETURN OLDVALUE]))])
```

(PLOTTICFN

; Edited 6-May-87 09:23 by jop

```
[LAMBDA ARGS
  (if (ILESSP ARGS 2)
    then (HELP "Must have at least two args"))
  (PROG ((PLOT (ARG ARGS 1))
      (AXIS (ARG ARGS 2))
      AXISINFO OLDVALUE)
    (SETQ AXISINFO (SELECTQ AXIS
      (X (fetch (PLOTSCALE XAXISINFO) of (fetch PLOTSCALE of PLOT)))
      (Y (fetch (PLOTSCALE YAXISINFO) of (fetch PLOTSCALE of PLOT)))
      (SHOULDNT)))
    (SETQ OLDVALUE (fetch (AXISINFO TICFN) of AXISINFO))
    (if (IGREATERP ARGS 2)
      then (LET [(NEWVALUE (ARG ARGS 3))
                (NODRAWFLG (AND (IGREATERP ARGS 3)
                              (ARG ARGS 4))
                    (replace (AXISINFO TICFN) of AXISINFO with NEWVALUE)
                    (RESCALEPLOT PLOT AXIS NODRAWFLG))]
      (RETURN OLDVALUE]))])
```

(PLOTTICINFO

; Edited 6-May-87 09:24 by jop

```
[LAMBDA (PLOT AXIS NEWTICINFO NODRAWFLG)
  (PROG ((PLOTSCALE (fetch PLOTSCALE of PLOT))
      OLDVALUE)
    (SETQ OLDVALUE (SELECTQ AXIS
      (X (fetch (PLOTSCALE XTICINFO) of PLOTSCALE))
      (Y (fetch (PLOTSCALE YTICINFO) of PLOTSCALE))
      (SHOULDNT)))
    (if (type? TICINFO NEWTICINFO)
      then (SELECTQ AXIS
        (X (replace (PLOTSCALE XTICINFO) of PLOTSCALE with NEWTICINFO))
        (Y (replace (PLOTSCALE YTICINFO) of PLOTSCALE with NEWTICINFO))
        (SHOULDNT))
      (if (NULL NODRAWFLG)
        then (REDRAWPLOTWINDOW PLOT)))
    (RETURN OLDVALUE]))
```

(PLOTTICMETHOD

; Edited 6-May-87 09:24 by jop

;; If NEWMETHOD not present then RETURNS current tic method for margin MARGIN , else replaces the method with NEWMETHOD, which may  
 ;; be a list of numbers, or a list of CONS pairs (VALUE . LABEL), or a function to be APPLIED to MARGIN PLOTSCALE PLOT, or the atom  
 ;; DEFAULT

```
(PROG (MARGIN OLDVALUE)
  (SETQ MARGIN (SELECTQ MARGINNAME
    (BOTTOM (fetch BOTTOMMARGIN of PLOT))
    (LEFT (fetch LEFTMARGIN of PLOT))
    (TOP (fetch TOPMARGIN of PLOT))
    (RIGHT (fetch RIGHTMARGIN of PLOT))
    (HELP "ILLEGAL MARGIN" MARGIN)))
  (SETQ OLDVALUE (fetch (MARGIN TICMETHOD) of MARGIN))
  (if NEWMETHOD
    then (replace (MARGIN TICMETHOD) of MARGIN with NEWMETHOD)
    (if (AND (NULL NODRAWFLG)
            (fetch TICS? of MARGIN))
      then (REDRAWPLOTWINDOW PLOT)))
  (RETURN OLDVALUE]))
```

(PLOTICS

; Edited 6-May-87 09:24 by jop

```
[LAMBDA ARGS
  (COND
```





```

(CAR MINSIZE))
(LESSP (WINDOWPROP PLOTWINDOW 'HEIGHT)
(CDR MINSIZE)))
(PROMPTPRINT "Plotwindow too small: reshape")
; Assumes SHAPEW will call REDRAWPLOTWINDOW
(SHAPEW PLOTWINDOW))
(T (ADJUSTVIEWPORT PLOTWINDOWVIEWPORT (DSPCLIPPINGREGION NIL PLOTWINDOW)
PLOT)
(CLEARW PLOTWINDOW)
(DRAWPLOT PLOT (WINDOWPROP PLOTWINDOW 'DSP)
PLOTWINDOWVIEWPORT
(DSPCLIPPINGREGION NIL PLOTWINDOW))
(COND
(SELECTEDOBJECT (HIGHLIGHTPLOTOBJECT SELECTEDOBJECT PLOT]))

```

(RELABELSELECTEDPLOTOBJECT

```

[LAMBDA (SELECTEDOBJECT PLOT) ; Edited 6-May-87 09:26 by jop
(PROG ((PLOTWINDOW (fetch (PLOT PLOTWINDOW) of PLOT))
LABEL LABELFLG) ; If the object is labeled, delete the label.
(if (PLOTOBJECTPROP SELECTEDOBJECT 'LABEL)
then (UNLABELPLOTOBJECT SELECTEDOBJECT PLOT)
(setq LABELFLG T))
(SETQ LABEL (fetch (PLOTOBJECT OBJECTLABEL) of SELECTEDOBJECT))
(TERPRI PLOTWINDOW)
[SETQ LABEL (PROMPTFORWARD "TYPE NEW LABEL : " LABEL "ENTER NIL FOR NO LABEL" PLOTWINDOW NIL NIL
CHARCODE (EOL LF ESCAPE TAB)
(replace (PLOTOBJECT OBJECTLABEL) of SELECTEDOBJECT with LABEL)
(LABELPLOTOBJECT SELECTEDOBJECT PLOT))

```

(RESCALEPLOT

```

[LAMBDA (PLOT AXIS NODRAWFLG) ; Edited 6-May-87 09:26 by jop
[COND
((NULL AXIS)
(SETQ AXIS 'BOTH)
(LET* ((PLOTSCALE (fetch PLOTSCALE of PLOT))
(PLOTOBJECTS (fetch PLOTOBJECTS of PLOT))
(PLOTTEXTENT (EXTENTOFPLOT PLOT))
(MINX (fetch (EXTENT MINX) of PLOTTEXTENT))
(MAXX (fetch (EXTENT MAXX) of PLOTTEXTENT))
(MINY (fetch (EXTENT MINY) of PLOTTEXTENT))
(MAXY (fetch (EXTENT MAXY) of PLOTTEXTENT)))
(COND
(PLOTOBJECTS (LET ((XINTERVAL (fetch (PLOTSCALE XINTERVAL) of PLOTSCALE))
(XAXISINFO (fetch (PLOTSCALE XAXISINFO) of PLOTSCALE))
(YINTERVAL (fetch (PLOTSCALE YINTERVAL) of PLOTSCALE))
(YAXISINFO (fetch (PLOTSCALE YAXISINFO) of PLOTSCALE))
TEMP)
[COND
((AND (OR (EQ AXIS 'BOTH)
(EQ AXIS 'X))
(GREATERP MAXX MINX))
(LET ((XAXISINFO (fetch (PLOTSCALE XAXISINFO) of PLOTSCALE))
TICINFO)
(SETQ TICINFO (CHOOSE TICS MINX MAXX XAXISINFO PLOT))
(replace (PLOTSCALE XTICINFO) of PLOTSCALE with TICINFO)
(replace (PLOTSCALE XINTERVAL) of PLOTSCALE
with (CHOOSE SCALE MINX MAXX XAXISINFO TICINFO PLOT))
[COND
((AND (OR (EQ AXIS 'BOTH)
(EQ AXIS 'Y))
(GREATERP MAXY MINY))
(LET ((YAXISINFO (fetch (PLOTSCALE YAXISINFO) of PLOTSCALE))
TICINFO)
(SETQ TICINFO (CHOOSE TICS MINY MAXY YAXISINFO PLOT))
(replace (PLOTSCALE YTICINFO) of PLOTSCALE with TICINFO)
(replace (PLOTSCALE YINTERVAL) of PLOTSCALE
with (CHOOSE SCALE MINY MAXY YAXISINFO TICINFO PLOT))
(COND
((NULL NODRAWFLG)
(REDRAWPLOTWINDOW PLOT))

```

(SCALE

```

[LAMBDA (MIN MAX NTICS ROUND POWER) ; Edited 6-May-87 09:26 by jop
;; Scaling algorithm for plots. NTICS is the desired number of tics. Round is a list of acceptable scaling factors. POWER is the power of ten to
;; use. Returns a TICINFO including NEWMAX, NEWMIN, INC, and NTICS
[COND
((NULL ROUND)
(SETQ ROUND '(5.0 2.5 2.0 1.5 1.0)) ; Rounding Constants. Notice that they are in decreasing order
; and end with 1.0
(PROG ((NUMINC (SUB1 NTICS))
RAWINC MANTISSA INDEX)
(SETQ RAWINC (FQUOTIENT (DIFFERENCE MAX MIN)
NUMINC)) ; POWER is the power of ten

```

```
[SETQ POWER (EXPT 10.0 (OR POWER (PLOT.FLOOR (PLOT.LOG10 RAWINC)
; MANTISSA is the scale factor
(SETQ MANTISSA (FQUOTIENT RAWINC POWER))
[COND
((GREATERP MANTISSA (CAR ROUND))
(SETQ POWER (TIMES 10 POWER))
(SETQ INDEX (LAST ROUND)))
(T (SETQ INDEX (for MARK on ROUND as TEST in (CDR ROUND) until (GREATERP MANTISSA TEST)
finally (RETURN MARK])
```

:: Find new max and new min

```
(RETURN (bind (NEWMAX _ MIN)
NEWMIN INC FACTOR LOWERMULT UPPERMULT while (LESSP NEWMAX MAX)
do (SETQ INC (TIMES (CAR INDEX)
POWER))
(SETQ FACTOR (FQUOTIENT (FDIFFERENCE (FPLUS MAX MIN)
(TIMES NUMINC INC))
(FTIMES 2.0 INC)))
[SETQ NEWMIN (FTIMES INC (SETQ LOWERMULT (PLOT.CEILING FACTOR)
[COND
((GREATERP NEWMIN MIN)
(SETQ NEWMIN (FTIMES INC (SETQ LOWERMULT (SUB1 LOWERMULT)
(COND
((AND (GEQ MIN 0.0)
(MINUSP NEWMIN))
(SETQ LOWERMULT 0)
(SETQ NEWMIN 0.0)))
(SETQ UPPERMULT (IPLUS LOWERMULT NUMINC))
(SETQ NEWMAX (FTIMES INC UPPERMULT))
[COND
((AND (LEQ MAX 0.0)
(GREATERP NEWMAX 0.0))
(SETQ UPPERMULT 0)
(SETQ NEWMAX 0.0)
(SETQ LOWERMULT (IMINUS NUMINC))
(SETQ NEWMIN (SETQ NEWMIN (FTIMES INC LOWERMULT)
[COND
((NULL (SETQ INDEX (NLEFT ROUND 1 INDEX)))
(SETQ INDEX (LAST ROUND))
(SETQ POWER (TIMES 10 POWER)
finally (RETURN (create TICINFO
TICMAX _ NEWMAX
TICMIN _ NEWMIN
TICINC _ INC
NTICS _ NTICS))
```

(TOGGELABEL

```
[LAMBDA (SELECTEDOBJECT PLOT) ; Edited 6-May-87 09:26 by jop
(COND
((PLOTOBJECTPROP SELECTEDOBJECT 'LABEL)
(UNLABELPLOT OBJECT SELECTEDOBJECT PLOT))
(T (LABELPLOT OBJECT SELECTEDOBJECT PLOT))
```

(TOGGLEEXTENDEDAXES

```
[LAMBDA (PLOT AXIS) (* jop%: "10-Dec-85 17:56")
```

(\*\*)

```
[COND
((NULL AXIS)
(SETQ AXIS 'BOTH]
[PROG [(XSCALEFN (PLOTSCALEFN PLOT 'X))
(YSCALEFN (PLOTSCALEFN PLOT 'Y))
[COND
((OR (EQ AXIS 'X)
(EQ AXIS 'BOTH))
(COND
((EQ XSCALEFN (FUNCTION EXTENDEDSCALEFN)) (* recover previous state)
(PLOTSCALEFN PLOT 'X (PLOTPROP PLOT 'OLDXSCALEFN)
T))
(T (* Remember the old fn for next time)
(PLOTPROP PLOT 'OLDXSCALEFN (PLOTSCALEFN PLOT 'X))
(PLOTSCALEFN PLOT 'X (FUNCTION EXTENDEDSCALEFN)
T]
(COND
((OR (EQ AXIS 'Y)
(EQ AXIS 'BOTH))
(COND
((EQ YSCALEFN (FUNCTION EXTENDEDSCALEFN))
(PLOTSCALEFN PLOT 'Y (PLOTPROP PLOT 'OLDYSCALEFN)
T))
(T (PLOTPROP PLOT 'OLDYSCALEFN (PLOTSCALEFN PLOT 'Y))
(PLOTSCALEFN PLOT 'Y (FUNCTION EXTENDEDSCALEFN)
T]
(RESCALEPLOT PLOT AXIS])
```

(TOGGLEFIXEDMENU

[LAMBDA (PLOT)

(\* jop%: "12-Dec-85 10:34")

(\* \*)

(PLOT.FIXRIGHTMENU PLOT (NOT (PLOT.FIXRIGHTMENU PLOT))

(TOGGLETICS

[LAMBDA (PLOT MARGINNAME)

(\* jop%: "10-Dec-85 21:27")

[COND

[(NULL MARGINNAME)

(for MARGIN in '(BOTTOM LEFT) do (COND

((PLOTTICS PLOT MARGIN)

(PLOTTICS PLOT MARGIN NIL T))

(T (PLOTTICS PLOT MARGIN T T]

(T (COND

((PLOTTICS PLOT MARGINNAME)

(PLOTTICS PLOT MARGINNAME NIL T))

(T (PLOTTICS PLOT MARGINNAME T T]

(REDRAWPLOTWINDOW PLOT])

(TRANSLATEPLOT OBJECT

[LAMBDA (OBJECT DX DY PLOT NODRAWFLG)

; Edited 6-May-87 09:27 by jop

(PROG [(TEXT OBJECT (PLOT OBJECTPROP OBJECT 'LABEL))

(WHENTRANSLATEDFN (PLOT OBJECTPROP OBJECT 'WHENTRANSLATEDFN]

(if (NULL NODRAWFLG)

then (if (EQ OBJECT (fetch (PLOT SELECTEDOBJECT) of PLOT))

then (LOWLIGHTPLOT OBJECT (fetch (PLOT SELECTEDOBJECT) of PLOT)

PLOT)

(replace (PLOT SELECTEDOBJECT) of PLOT with NIL))

(ERASEPLOT OBJECT PLOT))

; Destructively modify the data structure for OBJECT

(MOVEPLOT OBJECT DX DY PLOT)

(if (NULL NODRAWFLG)

then (DRAWPLOT OBJECT (fetch (PLOT PLOTWINDOWVIEWPORT) of PLOT)

PLOT))

(if TEXT OBJECT

then (TRANSLATEPLOT OBJECT TEXT OBJECT DX DY PLOT NODRAWFLG))

(APPLY.AFTERFN WHENTRANSLATEDFN OBJECT DX DY PLOT NODRAWFLG])

(UNDELETEPLOT OBJECT

[LAMBDA (PLOT MODE)

; Edited 6-May-87 09:27 by jop

;; MODE MAY BE ONE OF TOP, SELECT, ABOVE, ALL,.. NIL defaults to TOP. TOP means restore the top element of the save stack. SELECT means choose an object to restore from a menu. ABOVE means restore all objects above the selected object. ALL means restore all the objects on the save stack.

(if (NULL MODE) then (SETQ MODE 'TOP))

(PROG ((SAVELIST (fetch (PLOT PLOTSAVELIST) of PLOT))

SELECTION OBJECTSTORESTORE)

(if (NULL SAVELIST)

then (PLOT PROMPT "No object to undelete" PLOT)

(RETURN NIL))

(SETQ OBJECTSTORESTORE

(SELECTQ MODE

(TOP (LIST (CAR SAVELIST)))

(ALL SAVELIST)

((ABOVE SELECT)

[SETQ SELECTION (MENU (create MENU

ITEMS -

(bind OBJECTLABEL for OBJECT in SAVELIST as I from 1

collect (SETQ OBJECTLABEL (fetch (PLOT OBJECT OBJECTLABEL)

of OBJECT))

(LIST (if OBJECTLABEL

then (CONCAT (PLOT OBJECTSUBTYPE OBJECT)

" " OBJECTLABEL)

else (PLOT OBJECTSUBTYPE OBJECT))

I]

(AND SELECTION (if (EQ MODE 'SELECT)

then (LIST (CAR (NTH SAVELIST SELECTION)))

else (for I from 1 to SELECTION as OBJECT in SAVELIST collect OBJECT))))

(SHOULDNT "Illegal mode"))

[if OBJECTSTORESTORE

then (for OBJECT in OBJECTSTORESTORE do (ADDPLOT OBJECT PLOT))

(replace (PLOT PLOTSAVELIST) of PLOT with (SELECTQ MODE

(TOP (CDR SAVELIST))

(ALL NIL)

(ABOVE (CDR (NTH SAVELIST SELECTION)))

(SELECT (DREMOVE (CAR OBJECTSTORESTORE)

SAVELIST))

(SHOULDNT "ILLEGAL MODE"]

(RETURN OBJECTSTORESTORE])

**(UNLABELPLOT OBJECT)**

; Edited 6-May-87 09:27 by jop

```

[LAMBDA (OBJECT PLOT)

  (**)

  (PROG [(TEXTOBJECT (PLOT OBJECTPROP OBJECT 'LABEL))
        (WHENUNLABELED FN (PLOT OBJECTPROP OBJECT 'WHENUNLABELED FN))
        (COND
          (TEXTOBJECT (ERASEPLOT OBJECT TEXTOBJECT PLOT)
            (PLOT OBJECTPROP OBJECT 'LABEL NIL)
            (APPLY AFTERFN WHENUNLABELED FN OBJECT PLOT))
          (T (PLOT PROMPT "NOT A LABELED OBJECT" PLOT))])])

```

**(WHICH LABEL)**

; Edited 6-May-87 09:27 by jop

```

[LAMBDA (PLOT)

  ;; Prompt for new label and make the required call to ASKFOR LABEL

  (PROG ([LMENU (CONSTANT (create MENU
                          ITEMS _ ' (TOP LEFT BOTTOM RIGHT)
                          MARGIN)
          (PLOT PROMPT "Select a margin" PLOT)
          (SETQ MARGIN (MENU LMENU))
          (AND MARGIN (ASKFOR LABEL PLOT MARGIN))])])

```

**(WHICH PLOT)**

; Edited 6-May-87 09:27 by jop

```

[LAMBDA (X Y)

  ;; like WHICHW but returns corresponding plot. First arg may be a window

  (PROG ((W (OR (WINDOWP X)
                (WHICHW X Y)))
        (PLOT)
        [SETQ PLOT (OR (WINDOWPROP W 'PLOT)
                      (WINDOWPROP (WINDOWPROP W 'ICONFOR)
                                   'PLOT))
        (RETURN (COND
                  ((type? PLOT PLOT)
                   PLOT)
                  (T
                   PLOT)))))

```

;; Fns to do our own number printing

(DEFINEQ

**(PLOT.PRINTNUM)**

; Edited 7-May-87 17:23 by jop

```

[LAMBDA (F)

  (SETQ F (FLOAT F))
  (LET ((STR (CL:MAKE-ARRAY 14 :ELEMENT-TYPE 'CL:STRING-CHAR :FILL-POINTER 0))
        [MINUSFLAG (AND (< F 0.0)
                        (SETQ F (- F)
                                (ROUND 5)
                                NUMSTR INTEXP)
                        (IF (AND (OR (< F 0.001)
                                   (>= F 1.0E+7))
                              (NOT (ZEROP F)))
                            THEN (CL:MULTIPLE-VALUE-SETQ (NUMSTR INTEXP)
                                                           (FLTSTR F ROUND))
                            ELSE (CL:MULTIPLE-VALUE-SETQ (NUMSTR INTEXP)
                                                           (FLTSTR F ROUND))
                            (PLOT.FNUM-STRING STR NUMSTR INTEXP MINUSFLAG))])])

```

**(PLOT.FNUM-STRING)**

; Edited 7-May-87 17:21 by jop

```

[LAMBDA (OUTSTR MANTSTR INTEXP MINUSP)

  (LET* ((DIGITS (CL:LENGTH MANTSTR))
         (POINTPLACE (+ DIGITS INTEXP))
         (INDEX 0))

    (COND
      (MINUSP (CL:SETF (CL:AREF OUTSTR 0)
                      #\-)
              (SETQ INDEX 1)))

    [COND
      [((< POINTPLACE 0)
        (CL:SETF (CL:AREF OUTSTR INDEX)
                  #\0)
        (SETQ INDEX (CL:1+ INDEX))
        (CL:SETF (CL:AREF OUTSTR INDEX)
                  #\.)
        (SETQ INDEX (CL:1+ INDEX))
        (CL:DOTIMES (I (- POINTPLACE))
                    (CL:SETF (CL:AREF OUTSTR INDEX)
                              #\0)
                    (SETQ INDEX (CL:1+ INDEX)))]])

```

```

(CL:DOTIMES (I DIGITS)
  (CL:SETF (CL:AREF OUTSTR INDEX)
    (CL:AREF MANTSTR I))
  (SETQ INDEX (CL:1+ INDEX))))]
[(< INTEXP 0)
 (CL:DOTIMES (I POINTPLACE)
  (CL:SETF (CL:AREF OUTSTR INDEX)
    (CL:AREF MANTSTR I))
  (SETQ INDEX (CL:1+ INDEX)))
 (CL:SETF (CL:AREF OUTSTR INDEX)
  #\.)
 (SETQ INDEX (CL:1+ INDEX))
 (CL:DO ((I POINTPLACE (CL:1+ I))
  ((EQ I DIGITS))
  (CL:SETF (CL:AREF OUTSTR INDEX)
    (CL:AREF MANTSTR I))
  (SETQ INDEX (CL:1+ INDEX))))]
(T (CL:DOTIMES (I DIGITS)
  (CL:SETF (CL:AREF OUTSTR INDEX)
    (CL:AREF MANTSTR I))
  (SETQ INDEX (CL:1+ INDEX)))
 (CL:DOTIMES (I INTEXP)
  (CL:SETF (CL:AREF OUTSTR INDEX)
    #\0)
  (SETQ INDEX (CL:1+ INDEX)))
 (CL:SETF (CL:AREF OUTSTR INDEX)
  #\.)
 (SETQ INDEX (CL:1+ INDEX))
 (CL:SETF (CL:AREF OUTSTR INDEX)
  #\0)
 (SETQ INDEX (CL:1+ INDEX)
[COND
 ((OR (< POINTPLACE 0)
 (< INTEXP 0))
 ;; Trim off extraneous zeros
 (CL:DO ((I (CL:1- INDEX)
  (CL:1- I))
  [(NOT (EQ (CL:AREF OUTSTR I)
  #\0))
  (CL:IF (NOT (EQ (CL:AREF OUTSTR I)
  #\.)
  (SETQ INDEX (CL:1+ I))
  (SETQ INDEX (+ I 2))))]])
 (CL:SETF (CL:FILL-POINTER OUTSTR)
  INDEX)
OUTSTR])

```

**PLOT.ENUM-STRING**

; Edited 13-May-87 09:21 by jop

```

[LAMBDA (OUTSTR MANTSTR INTEXP MINUSP)
 ;; Prints exponential notation observing rounding & exponent spacing
 (LET ((DIGITS (CL:LENGTH MANTSTR))
  (INDEX 0)
  EXPOFFSET)
 (COND
  (MINUSP (CL:SETF (CL:AREF OUTSTR 0)
  #\-)
  (SETQ INDEX 1)))
 ;; Print the mantissa
 (CL:SETF (CL:AREF OUTSTR INDEX)
  (CL:AREF MANTSTR 0))
 (SETQ INDEX (CL:1+ INDEX))
 (CL:SETF (CL:AREF OUTSTR INDEX)
  #\.)
 (SETQ INDEX (CL:1+ INDEX))
 (CL:DO ((I 1 (CL:1+ I))
  ((EQ I DIGITS))
  (CL:SETF (CL:AREF OUTSTR INDEX)
    (CL:AREF MANTSTR I))
  (SETQ INDEX (CL:1+ INDEX))))
 ;; Trim off extraneous zeros
 (CL:DO ((I (CL:1- INDEX)
  (CL:1- I))
  [(NOT (EQ (CL:AREF OUTSTR I)
  #\0))
  (CL:IF (NOT (EQ (CL:AREF OUTSTR I)
  #\.)
  (SETQ INDEX (CL:1+ I))
  (SETQ INDEX (+ I 2))))]])
 ;; mantissa done - now for the exponent
 (SETQ EXPOFFSET (- (+ INTEXP DIGITS)
  1))

```

```
(SETQ MANTSTR (MKSTRING EXPOFFSET))
(SETQ DIGITS (CL:LENGTH MANTSTR))
(CL:SETF (CL:AREF OUTSTR INDEX)
  #\E)
(SETQ INDEX (CL:1+ INDEX))
(CL:DOTIMES (I DIGITS)
  (CL:SETF (CL:AREF OUTSTR INDEX)
    (CL:AREF MANTSTR I))
  (SETQ INDEX (CL:1+ INDEX)))
(CL:SETF (CL:FILL-POINTER OUTSTR)
  INDEX)
OUTSTR))
```

**(CREATETICLISTS**

```
[LAMBDA (PLOT) ; Edited 7-May-87 18:08 by jop
  (LET ((BOTTOMMARGIN (fetch (PLOT BOTTOMMARGIN) of PLOT))
        (LEFTMARGIN (fetch (PLOT LEFTMARGIN) of PLOT))
        (RIGHTMARGIN (fetch (PLOT RIGHTMARGIN) of PLOT))
        (TOPMARGIN (fetch (PLOT TOPMARGIN) of PLOT)))
    [IF (fetch (MARGIN TICS?) of BOTTOMMARGIN)
      THEN (replace (MARGIN TICLIST) of BOTTOMMARGIN with (NORMALIZE-TICLIST (GETTICLIST 'BOTTOM PLOT))]
    [IF (fetch (MARGIN TICS?) of LEFTMARGIN)
      THEN (replace (MARGIN TICLIST) of LEFTMARGIN with (NORMALIZE-TICLIST (GETTICLIST 'LEFT PLOT))]
    [IF (fetch (MARGIN TICS?) of RIGHTMARGIN)
      THEN (replace (MARGIN TICLIST) of RIGHTMARGIN with (NORMALIZE-TICLIST (GETTICLIST 'RIGHT PLOT))]
    [IF (fetch (MARGIN TICS?) of TOPMARGIN)
      THEN (replace (MARGIN TICLIST) of TOPMARGIN with (NORMALIZE-TICLIST (GETTICLIST 'TOP PLOT))]
    NIL]))
```

**(NORMALIZE-TICLIST**

```
[LAMBDA (TICLIST) ; Edited 27-May-87 18:19 by jop
  (BIND VALUE LABEL FOR TIC IN TICLIST COLLECT (IF (LISTP TIC)
    THEN (SETQ VALUE (CAR TIC))
          (SETQ LABEL (CDR TIC))
    ELSE (SETQ VALUE (SETQ LABEL TIC)))
    (CONS VALUE (IF (FLOATP LABEL)
      THEN (PLOT.PRINTNUM LABEL)
      ELSE LABEL]))
```

(DEFINEQ

**(DRAW-TICS-LEFT-RIGHT**

```
[LAMBDA (TICLIST MIN MAX RIGHTTIC LEFTTIC TICOFFSET TICFONT STREAM VIEWPORT LEFT-P) ; Edited 13-May-87 16:56 by jop
  (LET ((FONT (DSPFONT NIL STREAM)))
    (DSPFONT TICFONT STREAM)
    [bind YWINDOWLOC TICVALUE TICLABEL for TICPAIR in TICLIST
      do (SETQ TICVALUE (CAR TICPAIR))
          (SETQ TICLABEL (CDR TICPAIR))
          (if (AND (GEQ TICVALUE MIN)
                  (LEQ TICVALUE MAX))
              then (SETQ YWINDOWLOC (WORLDSTOSTREAMY TICVALUE VIEWPORT))
                  (MOVETO LEFTTIC YWINDOWLOC STREAM)
                  (DRAWTO RIGHTTIC YWINDOWLOC (DSPSCALE NIL STREAM)
                    'REPLACE STREAM)
                  (if TICLABEL
                      then (IF LEFT-P
                          THEN (MOVETO (DIFFERENCE LEFTTIC (PLUS TICOFFSET (STRINGWIDTH TICLABEL
                            STREAM))))
                          YWINDOWLOC STREAM)
                      ELSE (MOVETO (PLUS RIGHTTIC TICOFFSET)
                            YWINDOWLOC STREAM))
                  (PRIN1 TICLABEL STREAM])
    (DSPFONT FONT STREAM]))
```

**(DRAW-TICS-TOP-BOTTOM**

```
[LAMBDA (TICLIST MIN MAX TOPOFTIC BOTTOMOFTIC TICOFFSET TICFONT STREAM VIEWPORT BOTTOM-P) ; Edited 13-May-87 17:03 by jop
  (LET ((FONT (DSPFONT NIL STREAM)))
    (DSPFONT TICFONT STREAM)
    [bind XWINDOWLOC TICVALUE TICLABEL for TICPAIR in TICLIST
      do (SETQ TICVALUE (CAR TICPAIR))
          (SETQ TICLABEL (CDR TICPAIR))
          (if (AND (GEQ TICVALUE MIN)
                  (LEQ TICVALUE MAX))
              then (SETQ XWINDOWLOC (WORLDSTOSTREAMX TICVALUE VIEWPORT))
                  (MOVETO XWINDOWLOC TOPOFTIC STREAM)
                  (DRAWTO XWINDOWLOC BOTTOMOFTIC (DSPSCALE NIL STREAM)
                    'REPLACE STREAM)
                  (if TICLABEL
                      then (IF BOTTOM-P
                          ; always draw the tic mark
```



```

"Undelete last deleted
object")
(Select (UNDELETEPLOBJECT
'SELECT)
"Select object to undelete")
(Above (UNDELETEPLOBJECT
'ABOVE)
"Undelete all objects above
selected object")
(All (UNDELETEPLOBJECT 'ALL)
"Undelete all deleted objects")

```

```

)
(Fixed% Menu TOGGLEFIXEDMENU "Fix Plot menu"))

```

(RPAQQ **OBJECTOPSTABLE**

```

((POINT (DRAWFN DRAWPOINTOBJECT)
(ERASEFN ERASEPOINTOBJECT)
(HIGHLIGHTFN HIGHLIGHTPOINT)
(MOVEFN MOVEPOINT)
(LABELFN LABELPOINT)
(EXTENTFN EXTENTOFPOINT)
(DISTANCEFN DISTANCETOPOINT)
(COPYFN COPYPOINT)
(PUTFN PUTPOINT)
(GETFN GETPOINT))
(CURVE (DRAWFN DRAWCURVEOBJECT)
(ERASEFN ERASECURVEOBJECT)
(HIGHLIGHTFN HIGHLIGHTCURVE)
(MOVEFN MOVECURVE)
(EXTENTFN EXTENTOFCURVE)
(DISTANCEFN DISTANCETOCURVE)
(COPYFN COPYCURVE)
(PUTFN PUTCURVE)
(GETFN GETCURVE))
(POLYGON (DRAWFN DRAWPOLYGONOBJECT)
(ERASEFN ERASEPOLYGONOBJECT)
(HIGHLIGHTFN HIGHLIGHTPOLYGON)
(MOVEFN MOVEPOLYGON)
(EXTENTFN EXTENTOFPOLYGON)
(DISTANCEFN DISTANCETOPOLYGON)
(COPYFN COPYPOLYGON)
(PUTFN PUTPOLYGON)
(GETFN GETPOLYGON))
(LINE (DRAWFN DRAWLINEOBJECT)
(ERASEFN ERASELINEOBJECT)
(HIGHLIGHTFN HIGHLIGHTLINE)
(MOVEFN MOVELINE)
(EXTENTFN EXTENTOFFLINE)
(DISTANCEFN DISTANCETOLINE)
(COPYFN COPYLINE)
(PUTFN PUTLINE)
(GETFN GETLINE))
(GRAPH (DRAWFN DRAWGRAPHOBJECT)
(ERASEFN ERASEGRAPHOBJECT)
(HIGHLIGHTFN HIGHLIGHTGRAPH)
(EXTENTFN EXTENTOFGRAPH)
(DISTANCEFN DISTANCETOGRAPH)
(COPYFN COPYGRAPHOBJECT)
(PUTFN PUTGRAPH)
(GETFN GETGRAPH))
(TEXT (DRAWFN DRAWTEXTOBJECT)
(ERASEFN ERASETEXTOBJECT)
(HIGHLIGHTFN HIGHLIGHTTEXT)
(MOVEFN MOVETEXT)
(LABELFN LABELTEXT)
(EXTENTFN EXTENTOFTEXT)
(DISTANCEFN DISTANCETOTEXT)
(COPYFN COPYTEXT)
(PUTFN PUTTEXT)
(GETFN GETTEXT))
(COMPOUND (DRAWFN DRAWCOMPOUNDOBJECT)
(ERASEFN ERASECOMPOUNDOBJECT)
(HIGHLIGHTFN HIGHLIGHTCOMPOUND)
(LOWLIGHTFN LOWLIGHTCOMPOUND)
(MOVEFN MOVECOMPOUND)
(EXTENTFN EXTENTOFCOMPOUND)
(DISTANCEFN DISTANCETOCOMPOUND)
(COPYFN COPYCOMPOUND)
(PUTFN PUTCOMPOUND)
(GETFN GETCOMPOUND))
(FILLEDRECTANGLE (DRAWFN DRAWFILLEDRECTANGLEOBJECT)
(ERASEFN ERASEFILLEDRECTANGLEOBJECT)
(HIGHLIGHTFN HIGHLIGHTFILLEDRECTANGLE)
(MOVEFN MOVEFILLEDRECTANGLE)
(EXTENTFN EXTENTOFFILLEDRECTANGLE)
(DISTANCEFN DISTANCETOFFILLEDRECTANGLE)
(COPYFN COPYFILLEDRECTANGLE)

```



(PUTFN PUTFILLEDRECTANGLE)
(GETFN GETFILLEDRECTANGLE)))

(DECLARE%: EVAL@COMPILE

(DATATYPE EXTENT ((MINX FLOATING)
(MAXX FLOATING)
(MINY FLOATING)
(MAXY FLOATING)))

(DATATYPE MARGIN (TICS? TICMETHOD LABEL TICLIST))

[DATATYPE PLOT (PLOTOBJECTS PLOTSCALE SELECTEDOBJECT WINDOWINFO MARGININFO MENUINFO PLOTUSERDATA PLOTSAVELIST)
;; PLOTOBJECTS is a display list, PLOTSCALE describes the scale in world coordinates, USERDATA is a prop list, SAVELIST is for undelete
;; WINDOWINFO describes the associated PLOTWINDOW and its attached PLOTWINDOWVIEWPORT
(DATATYPE WINDOWINFO (PLOTWINDOW PLOTWINDOWVIEWPORT PLOTWINDOWVIEWPORT))
;; MARGININFO describes the size of the plot margins in stream coordinates
(DATATYPE MARGININFO (LEFTMARGIN RIGHTMARGIN TOPMARGIN BOTTOMMARGIN))
;; MENUINFO describes the PLOT's menus
(DATATYPE MENUINFO (MIDDLEMENU RIGHTMENU OTHERMENUS))
(ACCESSFNS PLOT ([XLOWER (fetch MIN of (fetch XINTERVAL of (fetch PLOTSCALE of DATUM)
[XUPPER (fetch MAX of (fetch XINTERVAL of (fetch PLOTSCALE of DATUM)
[YLOWER (fetch MIN of (fetch YINTERVAL of (fetch PLOTSCALE of DATUM)
[YUPPER (fetch MAX of (fetch YINTERVAL of (fetch PLOTSCALE of DATUM]

(DATATYPE PLOTFNS (DRAWFN ERASEFN HIGHLIGHTFN LOWLIGHTFN LABELFN MOVEFN EXTENTFN DISTANCEFN COPYFN PUTFN GETFN))

(DATATYPE PLOTOBJECT (OBJECTFNS OBJECTSUBTYPE OBJECTUSERDATA OBJECTMENU OBJECTLABEL OBJECTDATA))

(DATATYPE AXISINFO (SCALEFN TICFN) ; SCALEFN and TICFN are functions

)

[DATATYPE AXISINTERVAL ((MIN FLOATING)
(MAX FLOATING))
(ACCESSFNS (INTERVALLENGTH (FDIFFERENCE (fetch MAX of DATUM)
(fetch MIN of DATUM]

(DATATYPE PLOTSCALE (XINTERVAL XAXISINFO XTICINFO YINTERVAL YAXISINFO YTICINFO)
;; XINTERVAL YINTERVAL are instances of AXISINTERVAL, XAXISINFO and YAXISINFO are instances of AXISINFO and XTICINFO and
;; YTICINFO are instances of TICINFO

)

[DATATYPE TICINFO ((TICMIN FLOATING)
(TICMAX FLOATING)
TICINC NTICS)
(ACCESSFNS (TICINTERVALLENGTH (FDIFFERENCE (fetch (TICINFO TICMAX) of DATUM)
(fetch (TICINFO TICMIN) of DATUM]

)

(/DECLAREDATATYPE 'EXTENT ' (FLOATP FLOATP FLOATP FLOATP)
;; ---field descriptor list elided by lister---
' 8)

(/DECLAREDATATYPE 'MARGIN ' (POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 8)

(/DECLAREDATATYPE 'MENUINFO ' (POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 6)

(/DECLAREDATATYPE 'MARGININFO ' (POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 8)

(/DECLAREDATATYPE 'WINDOWINFO ' (POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 6)

(/DECLAREDATATYPE 'PLOT ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 16)

(/DECLAREDATATYPE 'PLOTFNS ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
POINTER)

```

;; ---field descriptor list elided by lister---
' 22)

(/DECLAREDATATYPE 'PLOT OBJECT ' (POINTER POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 12)

(/DECLAREDATATYPE 'AXISINFO ' (POINTER POINTER)
;; ---field descriptor list elided by lister---
' 4)

(/DECLAREDATATYPE 'AXISINTERVAL ' (FLOATP FLOATP)
;; ---field descriptor list elided by lister---
' 4)

(/DECLAREDATATYPE 'PLOTSCALE ' (POINTER POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 12)

(/DECLAREDATATYPE 'TICINFO ' (FLOATP FLOATP POINTER POINTER)
;; ---field descriptor list elided by lister---
' 8)

(DECLARE%: EVAL@COMPILE

(PUTPROPS APPLY.AFTERFN MACRO (ARGS (APPLY.AFTERFN.MACRO ARGS)))

(PUTPROPS PLOT OBJECTS SUBTYPE? MACRO [ARGS '(EQ ', (CAR ARGS)
(fetch (PLOT OBJECTS SUBTYPE) of , (CADR ARGS))]

(PUTPROPS PLOT OBJECT PROP MACRO (ARGS (PLOT OBJECT PROP MACRO ARGS)))

(PUTPROPS PLOT PROP MACRO (ARGS (PLOT PROP MACRO ARGS)))
)

(PUTPROPS PLOT OBJECT PROP ARG NAMES (NIL (PLOT OBJECT PROP NEW VALUE) . PROP ARGS))

(PUTPROPS PLOT.DEFAULT MENU ARG NAMES (NIL (MENU NAME NEW MENU ITEMS) . MENU ARGS))

(PUTPROPS PLOT.FIX RIGHT MENU ARG NAMES (NIL (PLOT FIXED FLG) . PROP ARGS))

(PUTPROPS PLOT LABEL ARG NAMES (NIL (PLOT MARGIN NAME NEW LABEL NODRAW FLG) . LABEL ARGS))

(PUTPROPS PLOT MENU ARG NAMES (NIL (PLOT MENU NAME NEW MENU) . MENU ARGS))

(PUTPROPS PLOT MENU ITEMS ARG NAMES (NIL (PLOT MENU NAME NEW MENU ITEMS) . MENU ARGS))

(PUTPROPS PLOT PRETTY FNS ARG NAMES (NIL (PLOT AXIS NEW PRETTY SCALE FN NEW INV PRETTY SCALE FN NODRAW FLG) . PROP ARGS))

(PUTPROPS PLOT PROP ARG NAMES (NIL (PLOT PROP NEW VALUE) . PROP ARGS))

(PUTPROPS PLOT SCALE FN ARG NAMES (NIL (PLOT AXIS NEW SCALE FN NODRAW FLG) . PROP ARGS))

(PUTPROPS PLOT TIC FN ARG NAMES (NIL (PLOT AXIS NEW TIC FN NODRAW FLG) . PROP ARGS))

(PUTPROPS PLOT TICS ARG NAMES (NIL (PLOT MARGIN NAME NEW TIC FLG NODRAW FLG) . LABEL ARGS))

(RPAQ? SMALL PLOT FONT ' (GACHA 8 MRR))

(RPAQ? LARGE PLOT FONT ' (GACHA 12 BRR))

```

;;; PLOT I/O

(DEFINEQ

**(COPY PLOT OBJECT**

[LAMBDA (PLOT OBJECT PLOT)

; Edited 5-May-87 18:26 by jop

```

;; Returns a copy of PLOT OBJECT. OBJECT PROPS are handled as follows. If the PLOT OBJECT has a COPY FN (which may be a list of fns) on
;; its prop list, apply's it to NEW PLOT OBJECT PLOT OBJECT PLOT and expects it to copy the OBJECT PROPS, else calls COPY ALL, except for
;; PLOT OBJECTS or lists of PLOT OBJECTS which are COPY OBJECT'ed

```

```

(PROG ([OBJECT COPY FN (MKLIST (PLOT OBJECT PROP PLOT OBJECT ' COPY FN)
NEW PLOT OBJECT)
(SETQ NEW PLOT OBJECT (CREATE PLOT OBJECT (fetch OBJECT FNS of PLOT OBJECT)
(PLOT OBJECTS SUBTYPE PLOT OBJECT)
(COPY ALL (fetch OBJECT LABEL of PLOT OBJECT))
(fetch OBJECT MENU of PLOT OBJECT)
(CL:FUNCALL (fetch (PLOT FNS COPY FN) of (fetch OBJECT FNS of PLOT OBJECT))
PLOT OBJECT PLOT)))

```

[for PROP NAME in (for PROP in (fetch OBJECT USER DATA of PLOT OBJECT) by (CDDR PROP) collect PROP)

```

do (PLOT OBJECTPROP NEWPLOT OBJECT PROPNAME
  (OR (AND OBJECTCOPYFN (bind PROPVALUE for FN in OBJECTCOPYFN
    until (SETQ PROPVALUE (CL:FUNCALL FN NEWPLOT OBJECT PLOT OBJECT
      PLOT PROPNAME))
    finally (RETURN PROPVALUE)))
  (LET ((PROPVALUE (PLOT OBJECTPROP PLOT OBJECT PROPNAME)))
    (COND
      ((type? PLOT OBJECT PROPVALUE)
       (COPYPLOT OBJECT PROPVALUE))
      [(LISTP PROPVALUE)
       (for ITEM in PROPVALUE collect (COND
         ((type? PLOT OBJECT ITEM)
          (COPYPLOT OBJECT ITEM PLOT))
         (T (HCOPYALL ITEM))
        )
      )
      (T (HCOPYALL PROPVALUE))
    )
  (COND
    ([OR (NOT (type? PLOT OBJECT NEWPLOT OBJECT))
     (NOT (EQ (PLOT OBJECTSUBTYPE NEWPLOT OBJECT)
              (PLOT OBJECTSUBTYPE PLOT OBJECT))
      (HELP "Not a plot object of correct type" NEWPLOT OBJECT)))
     (RETURN NEWPLOT OBJECT))
  )

```

(COPYPLOT

```

[LAMBDA (PLOT OPENFLG REGION TITLE BORDER) ; Edited 5-May-87 18:27 by jop
  ;; Copies a PLOT. Copying of PLOTPROP's is handled as follows. If PLOT has a COPYPLOTFN, (which may be a list of fns) calls it with
  ;; NEWPLOT PLOT as args, and expects it to copy the PLOTPROPS intelligently, else HCOPYALL's the PROPS, except for PLOT OBJECTS or
  ;; lists of PLOT OBJECTS which are COPY OBJECT'ed
  (PROG ([COPYFN (MKLIST (PLOTPROP PLOT 'COPYFN)
    (NEWPLOT (create PLOT))) ; OK to share Menus
    (replace (PLOT MIDDLEMENU) of NEWPLOT with (fetch (PLOT MIDDLEMENU) of PLOT))
    (replace (PLOT RIGHTMENU) of NEWPLOT with (fetch (PLOT RIGHTMENU) of PLOT))
    ; OTHERMENUS copied since it is a list in prop format and
    ; consists of MENU's or LITATOMS
    (replace (PLOT OTHERMENUS) of NEWPLOT with (COPY (fetch (PLOT OTHERMENUS) of PLOT)))
    (replace (PLOT LEFTMARGIN) of NEWPLOT with (create MARGIN copying (fetch (PLOT LEFTMARGIN) of PLOT)))
    (replace (PLOT RIGHTMARGIN) of NEWPLOT with (create MARGIN copying (fetch (PLOT RIGHTMARGIN) of PLOT)))
    (replace (PLOT TOPMARGIN) of NEWPLOT with (create MARGIN copying (fetch (PLOT TOPMARGIN) of PLOT)))
    (replace (PLOT BOTTOMMARGIN) of NEWPLOT with (create MARGIN copying (fetch (PLOT BOTTOMMARGIN) of PLOT)))
    ; Plot objects not shared since they can be destructively modified
    (replace (PLOT PLOT OBJECTS) of NEWPLOT with (for OBJECT in (fetch (PLOT PLOT OBJECTS) of PLOT)
      collect (COPYPLOT OBJECT OBJECT PLOT)))
    (replace (PLOT PLOTSCALE) of NEWPLOT with (create PLOTSCALE copying (fetch (PLOT PLOTSCALE) of PLOT)))
    ; Does a HCOPYALL since we don't know what's cached here
    [for PROPNAME in (for PROP in (fetch (PLOT PLOTUSERDATA) of PLOT) by (CDDR PROP) collect PROP)
      do (PLOTPROP NEWPLOT PROPNAME (OR (AND COPYFN (bind PROPVALUE for FN in COPYFN
        until (SETQ PROPVALUE (CL:FUNCALL FN NEWPLOT PLOT
          PROPNAME))
        finally (RETURN PROPVALUE)))
      (LET ((PROPVALUE (PLOTPROP PLOT PROPNAME)))
        (COND
          ((type? PLOT OBJECT PROPVALUE)
           (COPYPLOT OBJECT PROPVALUE))
          [(LISTP PROPVALUE)
           (for ITEM in PROPVALUE
            collect (COND
              ((type? PLOT OBJECT ITEM)
               (COPYPLOT OBJECT ITEM PLOT))
              (T (HCOPYALL ITEM))
            )
          )
          (T (HCOPYALL PROPVALUE))
        )
        ; Cache the display parameters
      )
    )
  (COND
    ((OR REGION TITLE BORDER)
     (replace (PLOT PLOTWINDOW) of NEWPLOT with (LIST REGION TITLE BORDER))
  )
  (COND
    (OPENFLG (OPENPLOTWINDOW NEWPLOT))
  )
  (RETURN NEWPLOT))

```

(PLOT OBJECTPRINT

```

[LAMBDA (PLOT OBJECT STREAM) ; Edited 7-May-87 10:27 by jop
  (PRINTOUT STREAM "#<" (fetch OBJECTSUBTYPE of PLOT OBJECT)
    " PLOT OBJECT>@")
  (\PRINTADDR PLOT OBJECT STREAM)
  T))

```

(PRINT PLOT OBJECT

```

[LAMBDA (PLOT OBJECT PLOT STREAM) ; Edited 5-May-87 18:27 by jop
  ;; Puts a plot object on STREAM
  (PROG [(OBJECTPUTFN (MKLIST (PLOT OBJECTPROP PLOT OBJECT 'PUTFN)
    (PRINTOUT STREAM "(READPLOT OBJECT) (" %, "OBJECTSUBTYPE" %, .P2 (fetch (PLOT OBJECT OBJECTSUBTYPE)
      of PLOT OBJECT)
      %, "OBJECTDATA" %, )
    (CL:FUNCALL (fetch (PLOT FNS PUTFN) of (fetch OBJECTFNS of PLOT OBJECT))

```

```

PLOT OBJECT PLOT STREAM)
(PRINTOUT STREAM %, "OBJECTMENU" %,)
(HPRINT (fetch OBJECTMENU of PLOT OBJECT)
  STREAM T T)
(PRINTOUT STREAM %, "OBJECTLABEL" %, .P2 (fetch OBJECTLABEL of PLOT OBJECT)
  %,)
(PRINTOUT STREAM "OBJECTUSERDATA (")
(for PROPNAME in (for PROP in (fetch OBJECTUSERDATA of PLOT OBJECT) by (CDDR PROP) collect PROP)
  do (PRINTOUT STREAM PROPNAME %,)
    (if (NULL (for FN in OBJECTPUTFN thereis (CL:FUNCALL FN PLOT OBJECT PLOT PROPNAME STREAM)))
      then (HPRINT (PLOT OBJECT PROP PLOT OBJECT PROPNAME)
        STREAM NIL T)))
(PRINTOUT STREAM ")))")
(RETURN T])

```

(PRINTPLOT

[LAMBDA (PLOT STREAM)

; Edited 5-May-87 18:27 by jop

:: Puts out a symbolic representation of PLOT on STREAM

```

(PROG ([PUTFN (MKLIST (PLOT PROP PLOT 'PUTFN)
  MENU)
(PRINTOUT STREAM " (READPLOT) (")
(PRINTOUT STREAM "RIGHTMENU" %,)
(if (EQ (PLOT.DEFAULTMENU 'RIGHT)
  (fetch (PLOT RIGHTMENU) of PLOT))
  then (PRINTOUT STREAM "DEFAULT" %,)
  else (HPRINT (fetch (PLOT RIGHTMENU) of PLOT)
    STREAM T T))
(PRINTOUT STREAM "MIDDLEMENU" %,)
(if (EQ (PLOT.DEFAULTMENU 'MIDDLE)
  (fetch (PLOT MIDDLEMENU) of PLOT))
  then (PRINTOUT STREAM "DEFAULT" %,)
  else (HPRINT (fetch (PLOT MIDDLEMENU) of PLOT)
    STREAM T T))
(for FIELDNAME in '( (PLOT OTHERMENUS)
  (PLOT LEFTMARGIN)
  (PLOT TOPMARGIN)
  (PLOT RIGHTMARGIN)
  (PLOT BOTTOMMARGIN)
  (PLOT PLOTSCALE))
  do (PRINTOUT STREAM (CADR FIELDNAME)
    %,)
  (HPRINT (RECORDACCESS FIELDNAME PLOT)
    STREAM T T))
(PRINTOUT STREAM %, "PLOT OBJECTS (")
(for OBJECT in (fetch (PLOT PLOT OBJECTS) of PLOT) do (HPRINT OBJECT STREAM T T))
(PRINTOUT STREAM ")" %,)
(PRINTOUT STREAM %, "PLOT USERDATA (")
(for PROPNAME in (for PROP in (fetch (PLOT PLOT USERDATA) of PLOT) by (CDDR PROP) collect PROP)
  do (PRINTOUT STREAM %, PROPNAME %,)
    (if (NULL (for FN in PUTFN thereis (CL:FUNCALL FN PLOT PROPNAME STREAM)))
      then (HPRINT (PLOT PROP PLOT PROPNAME)
        STREAM NIL T)))
(PRINTOUT STREAM ")" %,)
(PRINTOUT STREAM ")))")
(RETURN T])

```

(READFONT

[LAMBDA (STREAM)

(\* jop%: "27-Aug-85 13:34")

```

(PROG ((PROPLIST (READ STREAM)))
  (RETURN (FONTCREATE (LISTGET PROPLIST 'FAMILY)
    (LISTGET PROPLIST 'SIZE)
    (LISTGET PROPLIST 'FACE)
    (LISTGET PROPLIST 'ROTATION)
    (LISTGET PROPLIST 'DEVICE])))

```

(READMENU

[LAMBDA (STREAM)

; Edited 6-May-87 09:31 by jop

:: Function For Reading Menus From File

```

(PROG ((PROPLIST (HREAD STREAM)))
  (RETURN (create MENU
    ITEMS _ (LISTGET PROPLIST 'ITEMS)
    WHENSELECTEDFN _ (LISTGET PROPLIST 'WHENSELECTEDFN)
    WHENHELDFN _ (LISTGET PROPLIST 'WHENHELDFN)
    WHENUNHELDFN _ (LISTGET PROPLIST 'WHENUNHELDFN)
    MENUPOSITION _ (LISTGET PROPLIST 'MENUPOSITION)
    MENUOFFSET _ (LISTGET PROPLIST 'MENUOFFSET)
    MENUFONT _ (LISTGET PROPLIST 'MENUFONT)
    TITLE _ (LISTGET PROPLIST 'TITLE)
    CENTERFLG _ (LISTGET PROPLIST 'CENTERFLG)
    MENUROWS _ (LISTGET PROPLIST 'MENUROWS)
    MENCOLUMNS _ (LISTGET PROPLIST 'MENCOLUMNS)
    ITEMHEIGHT _ (LISTGET PROPLIST 'ITEMHEIGHT)

```

ITEMWIDTH \_ (LISTGET PROPLST 'ITEMWIDTH)
MENUBORDERSIZE \_ (LISTGET PROPLST 'MENUBORDERSIZE)
MENUOUTLINESIZE \_ (LISTGET PROPLST 'MENUOUTLINESIZE)
CHANGEOFFSETFLG \_ (LISTGET PROPLST 'CHANGEOFFSETFLG])

(READPLOT OBJECT

[LAMBDA (STREAM)

; Edited 5-May-87 18:27 by jop

:: Reads a plot object from STREAM previously written out by PRINTOBJECT

(PROG ((PROPLST (HREAD STREAM))
OBJECTSUBTYPE OBJECTFNS OBJECTGETFN NEWOBJECT OBJECTUSERDATA)
(SETQ OBJECTSUBTYPE (LISTGET PROPLST 'OBJECTSUBTYPE))
[SETQ OBJECTFNS (EVAL (PACK\* OBJECTSUBTYPE 'FNS))
(SETQ OBJECTGETFN (fetch (PLOTFNS GETFN) of OBJECTFNS))
[SETQ NEWOBJECT (CREATEPLOT OBJECTFNS OBJECTSUBTYPE (LISTGET PROPLST 'OBJECTLABEL)
(LISTGET PROPLST 'OBJECTMENU)
(CL:FUNCALL OBJECTGETFN (LISTGET PROPLST 'OBJECTDATA))
(SETQ OBJECTUSERDATA (LISTGET PROPLST 'OBJECTUSERDATA))
(for PROPNAME in OBJECTUSERDATA by (CDDR PROPNAME) as PROPVALUE in (CDR OBJECTUSERDATA)
by (CDDR PROPVALUE) do (PLOT OBJECTPROP NEWOBJECT PROPNAME (if (AND (LISTP PROPVALUE)
(EQ (CAR PROPVALUE)
'FUNCTION))
then (SETQ PROPVALUE
(CL:FUNCALL (CADR PROPVALUE)
NEWOBJECT PROPNAME))
else PROPVALUE)))
(RETURN NEWOBJECT)])

(READPLOT

[LAMBDA (STREAM)

; Edited 5-May-87 18:28 by jop

:: Reads In a Symbolic Representation Of A PLOT From Stream Previously Written Out By PRINTPLOT

(LET\* [(PROPLST (HREAD STREAM))
(RIGHTMENU (LISTGET PROPLST 'RIGHTMENU))
(MIDDLEMENU (LISTGET PROPLST 'MIDDLEMENU))
(USERDATA (LISTGET PROPLST 'PLOTUSERDATA))
(PLOT (create PLOT
OTHERMENUS \_ (LISTGET PROPLST 'OTHERMENUS)
LEFTMARGIN \_ (LISTGET PROPLST 'LEFTMARGIN)
TOPMARGIN \_ (LISTGET PROPLST 'TOPMARGIN)
RIGHTMARGIN \_ (LISTGET PROPLST 'RIGHTMARGIN)
BOTTOMMARGIN \_ (LISTGET PROPLST 'BOTTOMMARGIN)
PLOTSCALE \_ (LISTGET PROPLST 'PLOTSCALE)
PLOT OBJECTS \_ (LISTGET PROPLST 'PLOT OBJECTS])
(PLOTMENU PLOT 'RIGHT (if (EQ RIGHTMENU 'DEFAULT)
then (PLOT.DEFAULTMENU 'RIGHT)
else RIGHTMENU))
(PLOTMENU PLOT 'MIDDLE (if (EQ MIDDLEMENU 'DEFAULT)
then (PLOT.DEFAULTMENU 'MIDDLE)
else MIDDLEMENU))
(for PROPNAME in USERDATA by (CDDR PROPNAME) as PROPVALUE in (CDR USERDATA) by (CDDR PROPVALUE)
do (PLOTPROP PLOT PROPNAME (if (AND (LISTP PROPVALUE)
(AND (LISTP (CAR PROPVALUE))
(EQ (CAAR PROPVALUE)
'FUNCTION))
then ; Assumes Lists Of Form ((Function Foo) Bar)
(SETQ PROPVALUE (CL:FUNCALL (CADAR PROPVALUE)
PLOT PROPNAME (CADR PROPVALUE)))
else PROPVALUE)))
PLOT])

)

(DEFINEQ

(PRINT-VECTOR

[LAMBDA (VECTOR STREAM)

; Edited 1-Jun-87 17:34 by jop

(PRINTOUT STREAM "(READ-VECTOR)")
(PRIN2 (COERCE VECTOR 'LIST)
STREAM])

(READ-VECTOR

[LAMBDA (STREAM)

; Edited 1-Jun-87 17:39 by jop

(LET ((LST (HREAD STREAM))
(CL:MAKE-ARRAY (LENGTH LST)
:INITIAL-CONTENTS LST))

)

[PUTDEF 'PLOTS 'FILEPKGCOMS '((COM MACRO (PLTS (HORRIBLEVARS . PLTS)

(ADDTovar HPRINTMACROS (FONTDESCRIPTOR . PRINTFONT)
(MENU . PRINTMENU)
(PLOT . PRINTPLOT)

(PLOT OBJECT . PRINT PLOT OBJECT)  
(ONED-ARRAY . PRINT-VECTOR)

(ADDTOVAR HPRINTREADFNS READPLOT READPLOT OBJECT READFONT READMENU READ-VECTOR)

(DEFPRINT 'PLOT OBJECT (FUNCTION PLOT OBJECT PRINT))

;;; Numeric fns

(DEFINEQ

**(PLOT.EXP10**

[LAMBDA (X)

; Edited 6-May-87 09:32 by jop

;; this procedure returns exact power of ten for integer args

(EXPT 10.0 X)]

**(PLOT.LOG10**

[LAMBDA (X)

; Edited 6-May-87 09:32 by jop

;; Returns log base 10 of X

(PROG [(C (CONSTANT (FQUOTIENT 1.0 (LOG 10.0)  
(RETURN (FTIMES C (LOG X))

**(PLOT.FLOOR**

[LAMBDA (X)

; Edited 6-May-87 09:32 by jop

(SETQ X (FLOAT X))

(PROG ((FIXX (FIX X)))

(RETURN (COND

[ (MINUSP X)

(COND

((EQP FIXX X)

FIXX)

(T (SUB1 FIXX]

(T FIXX])

**(PLOT.CEILING**

[LAMBDA (X)

; Edited 6-May-87 09:32 by jop

(SETQ X (FLOAT X))

(PROG ((FIXX (FIX X)))

(RETURN (COND

((MINUSP X)

FIXX)

(T (COND

((EQP FIXX X)

FIXX)

(T (ADD1 FIXX])

**(SINEWAVE**

[LAMBDA (N FREQUENCY FROM TO AMPLITUDE)

; Edited 6-May-87 09:33 by jop

;; produce N points on a sine wave

(PROG ((TWOPI (TIMES 2.0 3.14159))

(RANGE (FDIFFERENCE TO FROM)))

(if (NULL FREQUENCY)

then (SETQ FREQUENCY 1))

(if (NULL AMPLITUDE)

then (SETQ AMPLITUDE 1))

(RETURN (bind (X \_ FROM)

(INC \_ (FQUOTIENT RANGE N))

POINT for I from 1 to N collect [SETQ POINT (create POSITION

XCOORD \_ X

YCOORD \_ (TIMES AMPLITUDE

(SIN (TIMES FREQUENCY X)

T]

(SETQ X (PLUS X INC))

POINT])

)

;;; PLOT image object FNS

(DEFINEQ

**(CREATEPLOTIMAGEOBJ**

[LAMBDA (PLOT)

; Edited 27-May-87 18:38 by jop

;; creates PLOT image object from PLOT

(LET\* ((WINDOW (fetch (PLOT PLOTWINDOW) of PLOT))

(REGION (IF (WINDOWP WINDOW)

THEN (WINDOWPROP WINDOW 'REGION)

```

ELSE (CAR WINDOW))
(OBJ (IMAGEOBJCREATE (COPYPLOT PLOT)
PLOTIMAGEFNS))
(IMAGEOBJPROP OBJ 'WIDTH (FETCH (REGION WIDTH) OF REGION))
(IMAGEOBJPROP OBJ 'HEIGHT (FETCH (REGION HEIGHT) OF REGION))
OBJ))

```

(CREATEPLOTBITMAPOBJ

```

[LAMBDA (PLOT) ; Edited 5-May-87 18:19 by jop
(LET* [(WINDOW (fetch (PLOT PLOTWINDOW) of PLOT))
(BITMAP (BITMAPCREATE (WINDOWPROP WINDOW 'WIDTH)
(WINDOWPROP WINDOW 'HEIGHT)
(BITBLT WINDOW NIL NIL BITMAP)
(BITMAPEDITOBJ BITMAP 1 0))

```

(PLIO.BUTTONEVENTINFN

```

[LAMBDA (PLOTIMAGEOBJ WINDOWSTREAM SELECTION RELX RELY WINDOW TEXTSTREAM BUTTON) ; Edited 6-May-87 09:34 by jop
(PROG ([CHOICEMENU (CONSTANT (create MENU
CENTERFLG _ T
ITEMS _ ' ("Select" 'SELECT "Select the image object")
("Reshape" 'RESHAPE "Reshape the image object")
("Plot Window" 'EDIT "Open a window containing plot"]
(PLOT (IMAGEOBJPROP PLOTIMAGEOBJ 'OBJECTDATUM))
(IMAGEWIDTH (IMAGEOBJPROP PLOTIMAGEOBJ 'WIDTH))
(IMAGEHEIGHT (IMAGEOBJPROP PLOTIMAGEOBJ 'HEIGHT))
MINSIZE NEWREGION WIN NEWPLOT)
;; consider selection if BUTTON=NIL to handle plots in Koto version of Sketch
(COND
((OR (NOT BUTTON)
(EQ BUTTON 'LEFT))
(SELECTQ (MENU CHOICEMENU)
(RESHAPE (SETQ MINSIZE (MINSTREAMREGIONSIZE (WINDOWPROP (fetch PLOTWINDOW of PLOT)
'DSP)
PLOT)) ; Assumes the WINDOWSTREAM has been changed to fit the
; imageobj
(SETQ NEWREGION (GETREGION (CAR MINSIZE)
(CDR MINSIZE)
(CREATEREGION (DSPXOFFSET NIL WINDOWSTREAM)
(DSPYOFFSET NIL WINDOWSTREAM)
IMAGewidth IMAGEHEIGHT)))
(IMAGEOBJPROP PLOTIMAGEOBJ 'WIDTH (fetch WIDTH of NEWREGION))
(IMAGEOBJPROP PLOTIMAGEOBJ 'HEIGHT (fetch HEIGHT of NEWREGION))
; Redraw the Image object
(RETURN 'CHANGED))
(EDIT (SETQ NEWPLOT (COPYPLOT PLOT NIL (GETBOXREGION (WIDTHIFWINDOW IMAGewidth)
(HEIGHTIFWINDOW IMAGEHEIGHT T))
"Plot Edit Window"))
(SETQ WIN (OPENPLOTWINDOW NEWPLOT))
;; Cache some info some that changes to NEWPLOT may be reinserted into TEXTSTREAM. Windowprops are used
;; because they are not copied (HACK)
;; sketch doesn't pass down anything for TEXTSTREAM arg so must use viewer window instead
(WINDOWPROP WIN 'SOURCEHOST (OR TEXTSTREAM WINDOW WINDOWSTREAM))
(WINDOWPROP WIN 'SOURCEIMAGEOBJ PLOTIMAGEOBJ)
(WINDOWADDPROP WIN 'CLOSEFN 'PLIO.EDITCLOSEFN T)
;; handle reinsert by a closefn rather than an new menu item -- similar to the behavior of Sketch image object edits
;; (PLOTADDMENUITEMS NEWPLOT (QUOTE RIGHT) (QUOTE ((Reinsert PLIO.REINSERTOBJ 'Change source
;; image object'))))
(RETURN T))
(RETURN NIL)))
(T (RETURN NIL])

```

(PLIO.COPYFN

```

[LAMBDA (PLOTIOBJ) ; Edited 6-May-87 09:35 by jop
; simple copy
(PROG ((NEWOBJ (IMAGEOBJCREATE NIL PLOTIMAGEFNS))
[IMAGEOBJPROP NEWOBJ 'OBJECTDATUM (COPYPLOT (IMAGEOBJPROP PLOTIOBJ 'OBJECTDATUM)
(IMAGEOBJPROP NEWOBJ 'WIDTH (IMAGEOBJPROP PLOTIOBJ 'WIDTH))
(IMAGEOBJPROP NEWOBJ 'HEIGHT (IMAGEOBJPROP PLOTIOBJ 'HEIGHT))
(RETURN NEWOBJ])

```

(PLIO.GETFN

```

[LAMBDA (STREAM TEXTSTREAM) ; Edited 6-May-87 09:35 by jop
;; PLOT IMAGEOBJECT GETFN
(PROG ((PROPLST (HREAD STREAM))
PLOTIMAGEOBJ)
(SETQ PLOTIMAGEOBJ (IMAGEOBJCREATE (LISTGET PROPLST 'PLOT)
PLOTIMAGEFNS))
(IMAGEOBJPROP PLOTIMAGEOBJ 'WIDTH (LISTGET PROPLST 'WIDTH))

```

```
(IMAGEOBJPROP PLOTIMAGEOBJ 'HEIGHT (LISTGET PROPLST 'HEIGHT))
(RETURN PLOTIMAGEOBJ)
```

**(PLIO.PUTFN**

[LAMBDA (PLOTIMAGEOBJ STREAM) ; Edited 6-May-87 09:35 by jop

```
;; PLOT IMAGEOBJECT PUTFN
(PRINTOUT STREAM "(WIDTH" %, (IMAGEOBJPROP PLOTIMAGEOBJ 'WIDTH)
%, "HEIGHT" %, (IMAGEOBJPROP PLOTIMAGEOBJ 'HEIGHT)
%, "PLOT" %,)
(HPRINT (IMAGEOBJPROP PLOTIMAGEOBJ 'OBJECTDATUM)
STREAM T T)
(PRINTOUT STREAM ")"])
```

**(PLIO.REINSERTOBJ**

[LAMBDA (PLOT) ; Edited 6-May-87 09:35 by jop

```
;; allows modified plot to be reinserted in document
;; modified to work with Sketch as well as TEdit sources
(PROG ((PLOTWINDOW (fetch PLOTWINDOW of PLOT))
HOST OBJ)
(SETQ HOST (WINDOWPROP PLOTWINDOW 'SOURCEHOST))
(SETQ OBJ (WINDOWPROP PLOTWINDOW 'SOURCEIMAGEOBJ))
(COND
((NOT (IMAGEOBJP OBJ))
(HELP "Not an IMAGEOBJ" OBJ))) ; Destructively change imageobj to retain EQ ness
(IMAGEOBJPROP OBJ 'OBJECTDATUM (COPYPLOT PLOT))
(IMAGEOBJPROP OBJ 'WIDTH (WINDOWPROP PLOTWINDOW 'WIDTH))
(IMAGEOBJPROP OBJ 'HEIGHT (WINDOWPROP PLOTWINDOW 'HEIGHT))
(IMAGE.OBJECT.CHANGED HOST OBJ))
```

**(PLOT.COPYBUTTONEVENTFN**

[LAMBDA (WINDOW) ; Edited 6-May-87 09:36 by jop

```
;; Allows plots to be copy selected
(PROG ((PLOT (WINDOWPROP WINDOW 'PLOT))
[IMAGETYPEMENU (CONSTANT (create MENU
ITEMS _ ' ((Plot 'PLOT)
(Bitmap 'BITMAP]
IMAGEOBJ)
(INVERTW WINDOW)
(UNTILMOUSESTATE UP)
(INVERTW WINDOW)
(COND
((INSIDEP WINDOW (CURSORPOSITION NIL WINDOW))
(SELECTQ (MENU IMAGETYPEMENU)
(PLOT (SETQ IMAGEOBJ (CREATEPLOTIMAGEOBJ PLOT)))
(BITMAP (SETQ IMAGEOBJ (CREATEPLOTBITMAPOBJ PLOT)))
NIL)
(AND IMAGEOBJ (COPYINSERT IMAGEOBJ))
```

**(PLIO.DISPLAYFN**

[LAMBDA (PLOTIOBJ IMAGESTREAM) ; Edited 7-May-87 18:21 by jop

```
;; Displays plot image object
(PROG ((PLOT (IMAGEOBJPROP PLOTIOBJ 'OBJECTDATUM))
(VIEWPORT (IMAGEOBJPROP PLOTIOBJ 'VIEWPORT))
(SCALE (DSPSCALE NIL IMAGESTREAM))
STREAMREGION)
(COND
((OR (NULL VIEWPORT)
(NOT (EQ (fetch PARENTSTREAM of VIEWPORT)
IMAGESTREAM)))
(SETQ VIEWPORT (CREATEVIEWPORT IMAGESTREAM))
(IMAGEOBJPROP PLOTIOBJ 'VIEWPORT VIEWPORT)))
[SETQ STREAMREGION (CREATEREGION (DSPXPOSITION NIL IMAGESTREAM)
(DSPYPOSITION NIL IMAGESTREAM)
[FIXR (TIMES SCALE (IMAGEOBJPROP PLOTIOBJ 'WIDTH]
[FIXR (TIMES SCALE (IMAGEOBJPROP PLOTIOBJ 'HEIGHT]
(CREATETICLISTS PLOT)
(ADJUSTVIEWPORT VIEWPORT STREAMREGION PLOT)
(DRAWPLOT PLOT IMAGESTREAM VIEWPORT STREAMREGION)]
```

**(PLIO.IMAGEBOXFN**

[LAMBDA (PLOTIOBJ IMAGESTREAM CURRENTX RIGHTMARGIN) ; Edited 6-May-87 09:36 by jop

```
;; Determines size of plotimageobj
(PROG ((IMAGEWIDTH (IMAGEOBJPROP PLOTIOBJ 'WIDTH))
(IMAGEHEIGHT (IMAGEOBJPROP PLOTIOBJ 'HEIGHT))
(PLOT (IMAGEOBJPROP PLOTIOBJ 'OBJECTDATUM))
(SCALE (COND
(IMAGESTREAM (DSPSCALE NIL IMAGESTREAM))
```



```

      (T 1)))
    NEWREGION MINSIZE)
;; (* this doesn't work with Sketch which has no rightmargin) (if (GREATERP (TIMES SCALE IMAGEWIDTH) (DIFFERENCE RIGHTMARGIN
;; CURRENTX) then (if (NOT (EQ (IMAGESTREAMTYPE IMAGESTREAM) (QUOTE DISPLAY))) then (HELP 'PLOT image object too big'))
;; (PROMPTPRINT 'Image object too wide. Choose a smaller region') (SETQ MINSIZE (MINSTREAMREGIONSIZEMINSTREAM PLOT))
;; (SETQ NEWREGION (GETREGION (CAR MINSIZE) (CDR MINSIZE))) (SETQ IMAGEWIDTH (fetch WIDTH of NEWREGION))
;; (IMAGEOBJPROP PLOTIOBJ (QUOTE WIDTH) IMAGEWIDTH) (SETQ IMAGEHEIGHT (fetch HEIGHT of NEWREGION)) (IMAGEOBJPROP
;; PLOTIOBJ (QUOTE HEIGHT) IMAGEHEIGHT))
    (RETURN (create IMAGEBOX
              XSIZE _ (TIMES SCALE IMAGEWIDTH)
              YSIZE _ (TIMES SCALE IMAGEHEIGHT)
              YDESC _ 0
              XKERN _ 0])
  )

```

;; additional fns to allow plot im. objs. to work in Sketch

(DEFINEQ

**(PLIO.EDITCLOSEFN**

[LAMBDA (W)

; Edited 5-May-87 18:10 by jop

;; this plot window is from an image object. Reinsert plot if requested  
;; later could test if plot has been changed -- if no changes don't ask to reinsert

```

  (LET (RESULT)
    (SETQ RESULT (SELECTQ (MENU (CONSTANT (create MENU
                                          TITLE _ "Change source image object?"
                                          ITEMS _ '("Yes" 'YES "This image used in the document
                                                    instead of the one that is there."
                                                    ("No" 'NO "The changes made to this image will
                                                    not be put into the document."))
                                          CENTERFLG _ T)))
          (YES (PLIO.REINSERTOBJ (WHICHPLOT W))
              NIL)
          (NO NIL)
          (NIL ; user selected outside the menu -- abort the close
              'DON'T)
          (NIL))
    (OR RESULT (WINDOWDELPROP W 'CLOSEFN 'PLIO.EDITCLOSEFN))
    RESULT))
; clean up window prop -- required since currently
; PLOT.CLOSEFN calls CLOSEW!

```

**(IMAGE.OBJECT.CHANGED**

[LAMBDA (HOST OBJECT)

; Edited 5-May-87 18:11 by jop

;; notifies HOST that OBJECT has changed and needs to be redisplayed  
;; currently assumes object is in TEdit or Sketch

```

  (LET (CANONICALHOST)
    (COND
      ([SETQ CANONICALHOST (CAR (NLSETQ (TEXTSTREAM HOST)
                                         (TEDIT.OBJECT.CHANGED CANONICALHOST OBJECT))
                                         (SETQ CANONICALHOST (CAR (NLSETQ (INSURE.SKETCH HOST)
                                         (SK.MARK.DIRTY CANONICALHOST)
                                         (for SKW in (SKETCH.ALL.VIEWERS CANONICALHOST) do (REDISPLAYW SKW)))
                                         (T (HELP "Can't update image object in " HOST))
                                         ; INSURE.SKETCH noerrorflg doesn't work
                                         ; this sets SKETCHCHANGED prop of all viewers on the sketch
    )

```

**(RPAQ? PLOTIMAGEFNS**

```

  (IMAGEFNSCREATE (FUNCTION PLIO.DISPLAYFN)
                  (FUNCTION PLIO.IMAGEBOXFN)
                  (FUNCTION PLIO.PUTFN)
                  (FUNCTION PLIO.GETFN)
                  (FUNCTION PLIO.COPYFN)
                  (FUNCTION PLIO.BUTTONEVENTINFN)
                  (FUNCTION NIL)
                  (FUNCTION NIL)
                  (FUNCTION NIL)
                  (FUNCTION NIL)
                  (FUNCTION NIL)
                  (FUNCTION NIL)))

```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS PLOTIMAGEFNS)

)

;;; Initialize

(PLOT.SETUP OBJECTTOPSTABLE)

(PLOT.DEFAULTMENU 'MIDDLE PLOT.DEFAULTMIDDLEMENUITEMS)

(PLOT.DEFAULTMENU 'RIGHT PLOT.DEFAULTRIGHTMENUITEMS)

;;; Dependent files

(FILESLOAD TWODGRAPHICS PLOTOBJECTS)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(FILESLOAD (LOADCOMP)  
TWODGRAPHICS UNBOXEDOPS)  
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)  
)  
)

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

(ADDTOVAR **NLAMA** )

(ADDTOVAR **NLAML** )

(ADDTOVAR **LAMA** PLOTTICS PLOTTICFN PLOTSCALEFN PLOTPROP PLOTOBJECTPROP PLOTMENUITEMS PLOTMENU PLOTLABEL  
PLOT.FIXRIGHTMENU PLOT.DEFAULTMENU)  
)

(PUTPROPS **PLOT COPYRIGHT** ("Xerox Corporation" 1985 1986 1987 1988 1993 2000))

FUNCTION INDEX

ADDPLOT OBJECT	2	EXTENT OF PLOT OBJECT	10	PLOTDELPROP	19
ADJUSTSCALE?	2	GETPLOTWINDOW	11	PLOT LABEL	19
ADJUSTVIEWPORT	3	GETTICLIST	11	PLOTMENU	19
APPLY.AFTERFN.MACRO	3	HIGHLIGHTPLOT OBJECT	11	PLOTMENUITEMS	20
ASKFOR LABEL	3	IMAGE.OBJECT.CHANGED	41	PLOT OBJECTADDPROP	20
ASKFORSCALE	3	LABELPLOT OBJECT	11	PLOT OBJECTDELPROP	20
BOXREGION	4	LOWLIGHTPLOT OBJECT	11	PLOT OBJECT LABEL	20
CHOOSES SCALE	4	MANUALRES SCALE	11	PLOT OBJECT PRINT	35
CHOOSE TICS	4	MINSTREAM REGION SIZE	12	PLOT OBJECT PROP	21
CLOSEPLOT WINDOW	4	MOVEPLOT OBJECT	12	PLOT OBJECT PROP MACRO	21
CLOSESTPLOT OBJECT	4	NORMALIZE-TICLIST	30	PLOT OBJECTS SUBTYPE	21
COMPOUND SUBTYPE	5	OPENPLOT WINDOW	12	PLOT OP ERROR	21
COMPUTE BOTTOM MARGIN	5	PLIO.BUTTONEVENTINFN	39	PLOT PROMPT	21
COMPUTE LEFT MARGIN	5	PLIO.COPYFN	39	PLOT PROP	22
COMPUTE RIGHT MARGIN	5	PLIO.DISPLAYFN	40	PLOT PROP MACRO	22
COMPUTE TOP MARGIN	6	PLIO.EDITCLOSEFN	41	PLOT REM PROP	22
COPYMENU	6	PLIO.GETFN	39	PLOTSCALEFN	22
COPYPLOT	35	PLIO.IMAGEBOXFN	40	PLOTTICFN	23
COPYPLOT OBJECT	34	PLIO.PUTFN	40	PLOTTICINFO	23
CREATEPLOT	6	PLIO.REINSERTOBJ	40	PLOTTICMETHOD	23
CREATEPLOTBITMAPOBJ	39	PLOT.BUTTONEVENTFN	13	PLOTTICS	23
CREATEPLOT FNS	7	PLOT.CEILING	38	PRINT-VECTOR	37
CREATEPLOT IMAGEOBJ	38	PLOT.CLOSEFN	14	PRINTFONT	24
CREATEPLOT OBJECT	7	PLOT.COPYBUTTONEVENTFN	40	PRINTMENU	24
CREATETICLISTS	30	PLOT.DEFAULTMENU	14	PRINTPLOT	36
DEFAULTSCALEFN	7	PLOT.ENUM-STRING	29	PRINTPLOT OBJECT	35
DEFAULTTICFN	7	PLOT.EXP10	38	READ-VECTOR	37
DEFAULTTICMETHOD	8	PLOT.FIXRIGHTMENU	15	READFONT	36
DELETEPLOT OBJECT	8	PLOT.FLOOR	38	READMENU	36
DESELECTPLOT OBJECT	8	PLOT.FNUM-STRING	28	READPLOT	37
DISTANCE TO PLOT OBJECT	8	PLOT.HARDCOPYFN	15	READPLOT OBJECT	37
DRAW-LABEL-LEFT-RIGHT	31	PLOT.ICONFN	16	REDRAWPLOT WINDOW	24
DRAW-LABEL-TOP-BOTTOM	31	PLOT.LABELTOWORLD	16	RELABELSELECTEDPLOT OBJECT	25
DRAW-TICS-LEFT-RIGHT	30	PLOT.LOG10	38	RESCALEPLOT	25
DRAW-TICS-TOP-BOTTOM	30	PLOT.PRINTNUM	28	SCALE	25
DRAWBOTTOMMARGIN	8	PLOT.REPAINTFN	17	SINEWAVE	38
DRAWLEFTMARGIN	9	PLOT.RESET	17	TOGGLABEL	26
DRAWMARGIN	9	PLOT.SETUP	17	TOGGLEEXTENDEDAXES	26
DRAWPLOT	9	PLOT.SKETCH.CREATE	17	TOGGLEFIXEDMENU	27
DRAWPLOT OBJECT	9	PLOT.WHENSELECTEDFN	17	TOGGLE TICS	27
DRAWRIGHTMARGIN	9	PLOT.WORLDTOLABEL	18	TRANSLATEPLOT OBJECT	27
DRAWTOPMARGIN	10	PLOTADDMENUITEMS	18	UNDELETEPLOT OBJECT	27
ERASEPLOT OBJECT	10	PLOTADDPROP	18	UNLABELPLOT OBJECT	28
EXTENDEDSCALEFN	10	PLOTAXISINTERVAL	18	WHICH LABEL	28
EXTENT OF PLOT	10	PLOTDELMENUITEMS	18	WHICH PLOT	28

PROPERTY INDEX

PLOT.DEFAULTMENU	34	PLOTMENU	34	PLOTPRETTYFNS	34	PLOTTICFN	34
PLOT.FIXRIGHTMENU	34	PLOTMENUITEMS	34	PLOTPROP	34	PLOTTICS	34
PLOT LABEL	34	PLOT OBJECT PROP	34	PLOTSCALEFN	34		

VARIABLE INDEX

HPRINTMACROS	37	OBJECTOPSTABLE	32	PLOTIMAGEFNS	41
HPRINTREADFNS	38	PLOT.DEFAULTMIDDLEMENUITEMS	31	SMALLPLOTFONT	34
LARGE PLOT FONT	34	PLOT.DEFAULTRIGHTMENUITEMS	31		

RECORD INDEX

AXISINFO	33	EXTENT	33	PLOT	33	PLOT OBJECT	33	TICINFO	33
AXISINTERVAL	33	MARGIN	33	PLOT FNS	33	PLOTSCALE	33		

MACRO INDEX

APPLY.AFTERFN	34	PLOT OBJECT PROP	34	PLOT OBJECTS SUBTYPE?	34	PLOT PROP	34
---------------	----	------------------	----	-----------------------	----	-----------	----