

File created: 15-Dec-86 11:35:38 {INDIGO}<GSLWS>KOTO>LIBRARY>MATHSERVERPLOT.;4

changes to: (FNS MAPL.Simple.MakePlot MAPL.ExpandFilename MAPL.Meta.MakePlot MAPL.Gen.MakePlot)  
(VARS MATHSERVERPLOTCOMS)

previous date: 8-Dec-86 09:46:39 {INDIGO}<GSLWS>KOTO>LIBRARY>MATHSERVERPLOT.;3

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(\* \* Copyright (c) 1986 by Xerox Corporation. All rights reserved.)

(RPAQQ **MATHSERVERPLOTCOMS**

```
(( * * ALL PLOTS)
(* Files for Log functions)
(FILESLoad plotexamples.dcom {INDIGO}<GSLWS>IDLPlot>FIXES>LOGPATCH.DCOM)
(* Low-level I/O functions)
(FNS MAPL.TopLevel MAPL.MakeReadtable MAPL.ReadASCIIline MAPL.ExpandFilename)
(* * METACODE PLOT STUFF)
(* Menu and window functions)
(FNS MAPL.Meta.TopLevel MAPL.Meta.FreeMenu MAPL.Meta.MakeIconWindow)
(* Plot functions)
(FNS MAPL.Meta.MakePlot MAPL.Meta.Plot MAPL.Meta.ASCIItoLisp)
(* Metaplot functions)
(FNS MAPL.Meta.NewPlotCom MAPL.Meta.MoveCom MAPL.Meta.DrawCom MAPL.Meta.EndPlotCom MAPL.Meta.NewPenCom)
(* Icon bitmaps)
(BITMAPS MAPL.Meta.Icon MAPL.Meta.IconMask)
(* * SIMPLE PLOT STUFF)
(* Menu and window functions)
(FNS MAPL.Simple.TopLevel MAPL.Simple.FreeMenu MAPL.Simple.MakeIconWindow)
(* Plot functions)
(FNS MAPL.Simple.MakePlot MAPL.Simple.Plot)
(* Icon bitmaps)
(BITMAPS MAPL.Simple.Icon MAPL.Simple.IconMask)
(* * GENERAL PLOT STUFF)
(* Menu and window functions)
(FNS MAPL.Gen.TopLevel MAPL.Gen.FreeMenu MAPL.Gen.MakeIconWindow)
(* Plot functions)
(FNS MAPL.Gen.MakePlot MAPL.Gen.Plot MAPL.Gen.ASCIItoLisp MAPL.Gen.NewPlot MAPL.Gen.PlotObject
  MAPL.Gen.EndPlot MAPL.Gen.NewPen MAPL.Gen.CollectData)
(* Icon bitmaps)
(BITMAPS MAPL.Gen.Icon MAPL.Gen.IconMask)
(* vars)
(P (MAPL.MakeReadtable))
(GLOBALVARS MAPL.ASCIIIRDTBL)
[ADDVARS (BackgroundMenuCommands (Plot% Menus (QUOTE (MAPL.TopLevel))
  "Opens all Plot Menus"
  (SUBITEMS (General% PlotMenu (QUOTE (MAPL.Gen.TopLevel))
    "Open a General Plot Menu")
    (Simple% PlotMenu (QUOTE (MAPL.Simple.TopLevel))
      "Open a Simple Plot Menu")
    (Meta% PlotMenu (QUOTE (MAPL.Meta.TopLevel))
      "Open a MetaCode Plot Menu"]
  (VARS (BackgroundMenu NIL))))))
```

(\* \* ALL PLOTS)

(\* \* Files for Log functions)

(FILESLoad plotexamples.dcom {INDIGO}<GSLWS>IDLPlot>FIXES>LOGPATCH.DCOM)

(\* \* Low-level I/O functions)

(DEFINEQ

(**MAPL.TopLevel**

[LAMBDA NIL

(\* DSB "5-Dec-86 11:39")

(\* opens all plot menus)

```
(MAPL.Gen.TopLevel)
(MAPL.Simple.TopLevel)
(MAPL.Meta.TopLevel])
```

(**MAPL.MakeReadtable**

[LAMBDA NIL

(\* DSB "24-Nov-86 09:53")

(\* \* Makes a readtable that reads ASCII records which end in carriage returns as strings, recognizing only CR as a separator character)

```
(SETQ MAPL.ASCIIIRDTBL (COPYREADTABLE FILERDTBL))
(SETSEPR (QUOTE (13))
  NIL MAPL.ASCIIIRDTBL])
```

**(MAPL.ReadASCIIline**

[LAMBDA (fileStream) (\* DSB "24-Nov-86 13:27")

(\* \* reads one record from a free-form ASCII file and returns a list of the data items in that record)

```
(LET ((stringstream (OPENSTRINGSTREAM (RSTRING fileStream MAPL.ASCIIRD_TBL))
    newChar)
    (READC fileStream)
    (while (NOT (EOFP stringstream)) collect (READ stringstream))
```

**(MAPL.ExpandFilename**

[LAMBDA (ITEM WINDOW BUTTONS) (\* DSB "15-Dec-86 11:21")  
(\* if filename is fullFilename, expand it into the separate slots)

```
(PROG ((state (FM.READSTATE WINDOW))
    filename host directory name extension version shortName)
  (SETQ filename (LISTGET state (QUOTE FILENAME)))
  (SETQ host (UNPACKFILENAME filename (QUOTE HOST)))
  [COND
    (host (SETQ directory (UNPACKFILENAME filename (QUOTE DIRECTORY)))
      (SETQ name (UNPACKFILENAME filename (QUOTE NAME)))
      (SETQ extension (UNPACKFILENAME filename (QUOTE EXTENSION)))
      (SETQ version (UNPACKFILENAME filename (QUOTE VERSION)))
      (SETQ shortName (PACKFILENAME (QUOTE NAME)
        name
        (QUOTE EXTENSION)
        extension
        (QUOTE VERSION)
        version))
      (FM.CHANGELABEL (FM.ITEMFROMID WINDOW (QUOTE FILENAME))
        WINDOW shortName)
      (FM.CHANGELABEL (FM.ITEMFROMID WINDOW (QUOTE DIRECTORY))
        WINDOW directory)
      (FM.CHANGELABEL (FM.ITEMFROMID WINDOW (QUOTE HOST))
        WINDOW host)
      (SETQ state (FM.READSTATE WINDOW))
    (RETURN state])
```

)

(\* \* METACODE PLOT STUFF)

(\* \* Menu and window functions)

(DEFINEQ

**(MAPL.Meta.TopLevel**

[LAMBDA NIL (\* DSB "5-Dec-86 11:49")  
(\* Sets up the MetaCode Plot Free Menu)

```
(PROG (menuWindow)
  (SETQ menuWindow (MAPL.Meta.FreeMenu))
  (FM.CHANGESTATE (FM.ITEMFROMID menuWindow (QUOTE PEN))
    menuWindow)
  (WINDOWPROP menuWindow (QUOTE ICONFN)
    (FUNCTION MAPL.Meta.MakeIconWindow))
  (SHAPEW menuWindow (QUOTE (200 420 271 127)))
  (OPENW menuWindow))
```

**(MAPL.Meta.FreeMenu**

[LAMBDA (LEFT BOTTOM) (\* DSB "3-Dec-86 12:33")  
(\* returns a free menu window for MetaCode plots at specified position)

```
(FM.FORMATMENU (BQUOTE ((TYPE TITLE LABEL Command: FONT (MODERN 12 BOLD))
  (LABEL MakePlot SELECTEDFN MAPL.Meta.MakePlot))
  ((TYPE TITLE LABEL "FILE INFO" FONT (MODERN 12 BOLD)))
  ((TYPE EDITSTART LABEL Filename: FONT (MODERN 12 BOLD))
    ITEMS
    (FILENAME))
  ((TYPE EDIT ID FILENAME LABEL ""))
  ((TYPE EDITSTART LABEL Directory: FONT (MODERN 12 BOLD))
    ITEMS
    (DIRECTORY))
  ((TYPE EDIT ID DIRECTORY LABEL ""))
  ((TYPE EDITSTART LABEL Host: FONT (MODERN 12 BOLD))
    ITEMS
    (HOST))
  ((TYPE EDIT ID HOST LABEL ""))
  ((TYPE TITLE LABEL "PLOT INFO" FONT (MODERN 12 BOLD)))
  ((TYPE TITLE LABEL PenWidth: FONT (MODERN 12 BOLD))
    (TYPE NWAY ID PEN LABEL ON)
    (TYPE NWAY ID PEN LABEL OFF))
  (WINDOWPROPS TITLE "MetaCode Plot Menu" LEFT , LEFT BOTTOM , BOTTOM]))
```

**(MAPL.Meta.MakelconWindow**

[LAMBDA (WINDOW OLDICON)

(\* DSB "5-Dec-86 18:01")

(\* \* Creates a shrink window with an icon formed by two bit maps.)

(OR OLDICON (ICONW MAPL.Meta.Icon MAPL.Meta.IconMask])

)

(\* \* Plot functions)

(DEFINEQ

**(MAPL.Meta.MakePlot**

[LAMBDA (ITEM WINDOW BUTTONS)

(\* DSB "15-Dec-86 11:32")

(\* checks that required data is specified and that the fullFilename is valid, and makes the MetaCode plot.)

(PROG ((promptW (GETPROMPTWINDOW WINDOW))  
state filename directory host penFlag fullFilename)

(\* \* check that all required data is specified)

```

(CLEARW promptW)
(SETQ state (MAPL.ExpandFilename ITEM WINDOW BUTTONS))
(SETQ filename (LISTGET state (QUOTE FILENAME)))
(COND
  ((EQUAL filename "")
   (PRIN1 "Unspecified file name." promptW)
   (RETURN)))
(SETQ directory (LISTGET state (QUOTE DIRECTORY)))
(COND
  ((EQUAL directory "")
   (PRIN1 "Unspecified directory." promptW)
   (RETURN)))
(SETQ host (LISTGET state (QUOTE HOST)))
(COND
  ((EQUAL host "")
   (PRIN1 "Unspecified host (DSK,IVY,etc.)" promptW)
   (RETURN)))
[SETQ penFlag (EQ (QUOTE ON)
                  (LISTGET state (QUOTE PEN]

```

(\* \* make fullFilename)

```

(SETQ fullFilename (PACKFILENAME (QUOTE HOST)
                                host
                                (QUOTE DIRECTORY)
                                directory
                                (QUOTE BODY)
                                filename))

```

(\* \* if fullFilename is valid, then make plot)

```

(COND
  ((NOT (INFILEP fullFilename))
   (PRIN1 "File not found" promptW)
   (RETURN))
  (T (PRIN1 "Making plot ..." promptW)
     (MAPL.Meta.Plot fullFilename penFlag)
     (CLEARW promptW)
     (PRIN1 "Done" promptW)
     (RETURN))

```

**(MAPL.Meta.Plot**

[LAMBDA (file penFlag)

(\* DSB "3-Dec-86 13:14")

(\* \* makes a PLOT of the metacode file)

```

(PROG (dataList plot code newVal1 newVal2 curveList pen)
  (SETQ dataList (MAPL.Meta.ASCIIToLisp file))
  [COND
    ((NOT dataList)
     (RETURN (PROMPTPRINT "There is no data")
              (* (PRIN1 dataList PROMPTWINDOW))
              ))
    (for item in dataList do
      (SETQ code (CAR item))
      (SETQ newVal1 (CADR item))
      (SETQ newVal2 (CADDR item))
      (SELECTQ code
        (1 (SETQ plot (MAPL.Meta.NewPlotCom))
            (* New plot)
            )
        (2 (SETQ curveList (MAPL.Meta.DrawCom curveList newVal1 newVal2)))
        (3 (SETQ curveList (MAPL.Meta.MoveCom plot curveList pen newVal1 newVal2))

```

)

```

(4 (MAPL.Meta.EndPlotCom plot curveList pen)
   (* End of plot)
)
(5 (SETQ pen (MAPL.Meta.NewPenCom newVal1 penFlag))
   (* New pen)
)
NIL])

```

**(MAPL.Meta.ASCIIToLisp**

[LAMBDA (file)

(\* DSB "25-Nov-86 09:54")

(\* returns ASCII data from a file in a list, with one sub-list per line.  
The file must end in a CR.)

```

(PROG ((tempFile (QUOTE {core}tempplot.dat))
      fileStream dataList)
  (COND
    ((NOT (INFILEP file))
     (RETURN NIL)))
  (COPYFILE file tempFile)

```

(\* copy to {core} because reads from filestream to {core} are much faster than reads from filestream to a VAX on the network.)

```

(SETQ fileStream (OPENSTREAM tempFile (QUOTE INPUT)))
(SETQ dataList (while (NOT (EOFP tempFile)) collect (MAPL.ReadASCIIline fileStream)))
(CLOSEF fileStream)
(DELFIL tempFile)
(RETURN dataList])

```

)

(\* \* Metaplot functions)

(DEFINEQ

**(MAPL.Meta.NewPlotCom**

[LAMBDA NIL

(\* DSB " 5-Dec-86 11:51")

(\* \* starts a new plot)

```

(PROG NIL
  (SETQ curveList NIL)
  (RETURN (CREATEPLOT NIL (QUOTE (471 420 250 250))
                    "MetaCode Plot"]))

```

**(MAPL.Meta.MoveCom**

[LAMBDA (plot curveList pen newVal1 newVal2)

(\* DSB "25-Nov-86 09:01")

(\* \* Plots the previous curve, and moves to a new position, starting a new curve)

```

(PROG NIL
  (COND
    ((AND curveList (GREATERP (LENGTH curveList)
                               1))
     (PLOT CURVE plot curveList NIL pen NIL T)))
  (SETQ curveList (LIST (CONS newVal1 newVal2)))
  (RETURN curveList])

```

**(MAPL.Meta.DrawCom**

[LAMBDA (curveList newVal1 newVal2)

(\* DSB "24-Nov-86 13:31")

(\* \* adds a new set of points to the curveList)

```

(PROG NIL
  (SETQ curveList (CONS (CONS newVal1 newVal2)
                        curveList))
  (RETURN curveList])

```

**(MAPL.Meta.EndPlotCom**

[LAMBDA (plot curveList pen)

(\* DSB "25-Nov-86 09:01")

(\* \* plots the last curve and opens the plot. It is expected that no more curves will be drawn to the plot)

```

(PROG NIL
  (COND
    ((AND curveList (GREATERP (LENGTH curveList)
                               1))
     (PLOT CURVE plot curveList NIL pen NIL T)))
  (OPENPLOTWINDOW plot])

```



```
(WINDOWPROP menuWindow (QUOTE ICONFN)
  (FUNCTION MAPL.Simple.MakeIconWindow))
(OPENW menuWindow])
```

**(MAPL.Simple.FreeMenu**

[LAMBDA (LEFT BOTTOM)

(\* DSB "8-Dec-86 07:59")

(\* returns a free menu window for simple plots at specified position)

```
(FM.FORMATMENU (BQUOTE ((TYPE TITLE LABEL Command: FONT (MODERN 12 BOLD))
  (LABEL MakePlot SELECTEDFN MAPL.Simple.MakePlot))
  ((TYPE TITLE LABEL "FILE INFO" FONT (MODERN 12 BOLD)))
  ((TYPE EDITSTART LABEL Filename: FONT (MODERN 12 BOLD)
    ITEMS
    (FILENAME))
  (TYPE EDIT ID FILENAME LABEL ""))
  ((TYPE EDITSTART LABEL Directory: FONT (MODERN 12 BOLD)
    ITEMS
    (DIRECTORY))
  (TYPE EDIT ID DIRECTORY LABEL ""))
  ((TYPE EDITSTART LABEL Host: FONT (MODERN 12 BOLD)
    ITEMS
    (HOST))
  (TYPE EDIT ID HOST LABEL ""))
  ((TYPE TITLE LABEL "PLOT INFO" FONT (MODERN 12 BOLD)))
  ((TYPE TITLE LABEL PlotType: FONT (MODERN 12 BOLD))
  (TYPE NWAY ID TYPE LABEL X-Y CLASSNAME XY)
  (TYPE NWAY ID TYPE LABEL X-LogY CLASSNAME X-LogY)
  (TYPE NWAY ID TYPE LABEL LogX-Y CLASSNAME LogX-Y)
  (TYPE NWAY ID TYPE LABEL LogX-LogY CLASSNAME LogX-LogY))
  ((TYPE TITLE LABEL Representation: FONT (MODERN 12 BOLD))
  (TYPE NWAY ID REP LABEL Curve CLASSNAME Curve)
  (TYPE NWAY ID REP LABEL Points CLASSNAME Points)
  (TYPE NWAY ID REP LABEL Both CLASSNAME Both))
  ((TYPE EDITSTART LABEL PenWidth: FONT (MODERN 12 BOLD)
    ITEMS
    (PEN))
  (TYPE EDIT ID PEN LABEL "1"))
  ((TYPE TITLE LABEL Destination: FONT (MODERN 12 BOLD))
  (TYPE NWAY ID DEST LABEL New CLASSNAME New)
  (TYPE NWAY ID DEST LABEL Previous CLASSNAME Previous)))
(WINDOWPROPS TITLE "Simple Plot Menu" LEFT , LEFT BOTTOM , BOTTOM])
```

**(MAPL.Simple.MakeIconWindow**

[LAMBDA (WINDOW OLDICON)

(\* DSB "5-Dec-86 18:00")

(\* \* Creates a window with an icon formed by two bit maps.)

```
(OR OLDICON (ICONW MAPL.Simple.Icon MAPL.Simple.IconMask])
```

)

(\* \* Plot functions)

(DEFINEQ

**(MAPL.Simple.MakePlot**

[LAMBDA (ITEM WINDOW BUTTONS)

(\* DSB "15-Dec-86 11:27")

(\* checks that required data is specified and that the fullFilename is valid, and makes the simple plot.)

```
(PROG ((promptW (GETPROMPTWINDOW WINDOW))
  state filename directory host type rep penWidth dest fullFilename)
```

(\* \* check that all required data is specified)

```
(CLEARW promptW)
(SETQ state (MAPL.ExpandFilename ITEM WINDOW BUTTONS))
(SETQ filename (LISTGET state (QUOTE FILENAME)))
(COND
  ((EQUAL filename "")
    (PRIN1 "Unspecified file name." promptW)
    (RETURN)))
(SETQ directory (LISTGET state (QUOTE DIRECTORY)))
(COND
  ((EQUAL directory "")
    (PRIN1 "Unspecified directory." promptW)
    (RETURN)))
(SETQ host (LISTGET state (QUOTE HOST)))
(COND
  ((EQUAL host "")
    (PRIN1 "Unspecified host (DSK,IVY,etc.)" promptW)
    (RETURN)))
(SETQ type (LISTGET state (QUOTE TYPE)))
(COND
```

```

    (NOT type)
    (PRIN1 "Unspecified plot type (XY, etc)." promptW)
    (RETURN))
  (SETQ rep (LISTGET state (QUOTE REP)))
  (COND
    ((NOT rep)
     (PRIN1 "Unspecified represent. (Points,etc.)" promptW)
     (RETURN))
    [SETQ penWidth (MKATOM (LISTGET state (QUOTE PEN)
  (COND
    ((AND (NUMBERP penWidth)
          (GREATERP penWidth 0))
     (SETQ penWidth (FIX penWidth)))
    (T (PRIN1 "PenWidth must be integer > 0" promptW)
      (RETURN)))
  (SETQ dest (LISTGET state (QUOTE DEST)))
  (COND
    ((NOT dest)
     (PRIN1 "Unspecified destination (New, etc.)" promptW)
     (RETURN)))

```

(\* \* make fullFilename)

```

  (SETQ fullFilename (PACKFILENAME (QUOTE HOST)
                                   host
                                   (QUOTE DIRECTORY)
                                   directory
                                   (QUOTE BODY)
                                   filename))

```

(\* \* if fullFilename is valid, then make plot)

```

  (COND
    ((NOT (INFILEP fullFilename))
     (PRIN1 "File not found" promptW)
     (RETURN))
    (T (PRIN1 "Making plot ..." promptW)
      (MAPL.Simple.Plot fullFilename promptW rep type penWidth dest)
      (CLEARW promptW)
      (PRIN1 "Done" promptW)
      (RETURN))

```

## (MAPL.Simple.Plot

[LAMBDA (filename promptW rep type penWidth dest)

(\* DSB " 8-Dec-86 08:53")

(\* Makes the plot and puts it into the appropriate window)

(\* \* takes an ASCII file of pairs of X-Y values, converts it into list format, and then converts it into a list of dotted pairs of data for requested plot.)

```

  (PROG ([rightMenuItems (QUOTE ((Logscale SCAT.LOGSCALE "Toggle exponential tics"
                                   (SUBITEMS (X% axis (SCAT.LOGSCALE (QUOTE X))
                                   "X axis only")
                                   (Y% axis (SCAT.LOGSCALE (QUOTE Y))
                                   "Y axis only"))))
        (Coordinates SCAT.WORLDCOORD "Display world coordinates at cursor position"]
        [pointMenuItems (QUOTE ((Coordinates SCAT.POINTCOORDS "Display point coordinates"]
        (tempFile (QUOTE {core}tempplot.dat))
        data first second mouseDown? fileStream dataList newPlot)

```

(\* \* copy to {core} and read into a list, with each line in the original file becoming a sub-list)

```

  (COPYFILE filename tempFile)
  (SETQ fileStream (OPENSTREAM tempFile (QUOTE INPUT)))
  (SETQ dataList (while (NOT (EOFP tempFile)) collect (MAPL.ReadASCIILine fileStream)))
  (CLOSEF fileStream)
  (DELFIL tempFile)

```

(\* \* translate to list of dotted pairs, depending on type of plot to be made)

```

  (SETQ data NIL)
  (COND
    [(EQUAL type (QUOTE X-Y))
     (for item in dataList do (SETQ first (CAR item))
                              (SETQ second (CADR item))
                              (COND
                                ((AND (NUMBERP first)
                                      (NUMBERP second))
                                 (SETQ data (CONS (CONS first second)
                                                    data))
                                (T (PRIN1 "Error: non-numeric data" promptW)
                                  (RETURN)))
     [(EQUAL type (QUOTE X-LogY))
     (for item in dataList do (SETQ first (CAR item))
                              (SETQ second (CADR item))
                              (COND
                                ((AND (NUMBERP first)
                                      (NUMBERP second)
                                      (GREATERP second 0))
                                 (SETQ data (CONS (CONS first second)
                                                    data))
                                (T (PRIN1 "Error: non-numeric data" promptW)
                                  (RETURN)))

```

```

                                (SETQ data (CONS (CONS first (PLOT.LOG10 second))
                                data]
[(EQUAL type (QUOTE LogX-Y))
 (for item in dataList do (SETQ first (CAR item))
                           (SETQ second (CADR item))
                           (COND
                            ((AND (NUMBERP first)
                                (GREATERP first 0)
                                (NUMBERP second))
                             (SETQ data (CONS (CONS (PLOT.LOG10 first)
                                                    second)
                                                data]
                             data]
                            (EQUAL type (QUOTE LogX-LogY))
                             (for item in dataList do (SETQ first (CAR item))
                                                         (SETQ second (CADR item))
                                                         (COND
                                                          ((AND (NUMBERP first)
                                                              (GREATERP first 0)
                                                              (NUMBERP second)
                                                              (GREATERP second 0))
                                                           (SETQ data (CONS (CONS (PLOT.LOG10 first)
                                                                 (PLOT.LOG10 second))
                                                                data]
                                                           data]
                                                          (T (CLEARW promptW)
                                                             (PRIN1 "Error: Unknown plot type" promptW)
                                                             (RETURN)))
                                                         (CLEARW promptW)

(* * If the new data is to be put on a previous plot, the user has 20 seconds to button in the desired plot window)

(* * otherwise, the data goes into a new plot)

[COND
 ((EQUAL dest (QUOTE Previous))
  (PRIN1 "Button in desired plot window" promptW)
  (SETQ mouseDown? (UNTILMOUSESTATE LEFT 20000))
  (COND
   [mouseDown? (COND
                 ((EQUAL rep (QUOTE Points))
                  (PLOTPOINTS (WHICHPLOT)
                               data)
                  (RETURN))
                 ((EQUAL rep (QUOTE Curve))
                  (PLOTCURVE (WHICHPLOT)
                              data NIL penWidth)
                  (RETURN))
                 ((EQUAL rep (QUOTE Both))
                  (PLOTPOINTS (WHICHPLOT)
                              data)
                  (PLOTCURVE (WHICHPLOT)
                              data NIL penWidth)
                  (RETURN))
                 (T (PRIN1 "Error: Unknown represent." promptW)
                    (RETURN))
                (T (PRIN1 "Making a new plot." promptW)
                   (SETQ newPlot (CREATEPLOT NIL (QUOTE (471 250 250 250))
                                             "Simple Plot"))
                   (PLOTADDMENUITEMS newPlot (QUOTE RIGHT)
                                         rightMenuItems)
                   (PLOTMENUITEMS newPlot (QUOTE POINTMENU)
                                         (APPEND (PLOTMENUITEMS newPlot (QUOTE MIDDLE))
                                                  pointMenuItems))
                   (COND
                    ((EQUAL rep (QUOTE Curve))
                     (PLOTCURVE newPlot data NIL penWidth))
                    ((EQUAL rep (QUOTE Points))
                     (PLOTPOINTS newPlot data))
                    ((EQUAL rep (QUOTE Both))
                     (PLOTCURVE newPlot data NIL penWidth)
                     (PLOTPOINTS newPlot data))
                    (T (PRIN1 "Error: Unknown represent." promptW)
                       (RETURN)))
                   (OPENPLOTWINDOW newPlot))
  )

(* * Icon bitmaps)

(RPAQ MAPL.Simple.Icon (READBITMAP))

```

```

(70 70 "0000000000000000L@@" "0000000000000000L@@" "L@@@@@@@@@ON@@@@@L@" "L@@@@@@@@@GO@@@@@L@"
 "L@@@@COOOOOOOOOOOL@@" "L@@@@B@@@@@AHO@@@@DL@" "L@@@@B@@@@@OL@@@@DL@" "L@@@@B@@@@@GN@@@@DL@"
 "L@@@@BB@@@@@CO@@@@DL@" "L@@@@BB@@@@@AHO@@@@DL@" "L@@@@BF@@@@@OHO@@@@DL@" "L@@@@BB@@@@@GL@@@@DL@"
 "L@@@@BB@@@@@CL@@@@DL@" "LOO@@@@B@@@@@AN@@@@DL@" "LLC@@@@BF@@@@@F@@@@DL@" "LLC@@@@B@@@@@C@@@@DL@"
 "LL@@@@BB@@@@@AH@@@@DL@" "LL@@@@BB@@@@@CL@@@@GH@@@@DL@" "LL@@@@BF@@@@@F@@@@AL@@@@DL@" "LLCNCBB@@@@LC@@@@DL@"

```



```
(FM.FORMATMENU (BQUOTE (( (TYPE TITLE LABEL Command: FONT (MODERN 12 BOLD))
(LABEL MakePlot SELECTEDFN MAPL.Gen.MakePlot))
((TYPE TITLE LABEL "FILE INFO" FONT (MODERN 12 BOLD)))
((TYPE EDITSTART LABEL Filename: FONT (MODERN 12 BOLD)
ITEMS
(FILENAME))
(TYPE EDIT ID FILENAME LABEL ""))
((TYPE EDITSTART LABEL Directory: FONT (MODERN 12 BOLD)
ITEMS
(DIRECTORY))
(TYPE EDIT ID DIRECTORY LABEL ""))
((TYPE EDITSTART LABEL Host: FONT (MODERN 12 BOLD)
ITEMS
(HOST))
(TYPE EDIT ID HOST LABEL ""))
((TYPE TITLE LABEL "PLOT INFO" FONT (MODERN 12 BOLD)))
((TYPE EDITSTART LABEL X-Position: FONT (MODERN 12 BOLD)
ITEMS
```

```

        (XPOS))
      (TYPE EDIT ID XPOS LABEL "1"))
    ((TYPE EDITSTART LABEL Y-Position: FONT (MODERN 12 BOLD)
      ITEMS
      (YPOS))
      (TYPE EDIT ID YPOS LABEL "2"))
    ((TYPE TITLE LABEL PlotType: FONT (MODERN 12 BOLD))
      (TYPE NWAY ID TYPE LABEL X-Y CLASSNAME XY)
      (TYPE NWAY ID TYPE LABEL X-LogY CLASSNAME X-LogY)
      (TYPE NWAY ID TYPE LABEL LogX-Y CLASSNAME LogX-Y)
      (TYPE NWAY ID TYPE LABEL LogX-LogY CLASSNAME LogX-LogY))
    ((TYPE TITLE LABEL Representation: FONT (MODERN 12 BOLD))
      (TYPE NWAY ID REP LABEL Curve CLASSNAME Curve)
      (TYPE NWAY ID REP LABEL Points CLASSNAME Points)
      (TYPE NWAY ID REP LABEL Both CLASSNAME Both))
    ((TYPE TITLE LABEL PenWidth: FONT (MODERN 12 BOLD))
      (TYPE NWAY ID PEN LABEL ON)
      (TYPE NWAY ID PEN LABEL OFF))
    (WINDOWPROPS TITLE "General Plot Menu" LEFT , LEFT BOTTOM , BOTTOM])

```

**(MAPL.Gen.MakelconWindow**

[LAMBDA (WINDOW OLDICON)

(\* DSB "5-Dec-86 17:42")

(\* \* Creates a shrink window with an icon formed by two bit maps.)

(OR OLDICON (ICONW MAPL.Gen.Icon MAPL.Gen.IconMask])

)

(\* \* Plot functions)

(DEFINEQ

**(MAPL.Gen.MakePlot**

[LAMBDA (ITEM WINDOW BUTTONS)

(\* DSB "15-Dec-86 11:30")

(\* checks that required data is specified and that the fullFilename is valid, and makes the simple plot.)

```

(PROG ((promptW (GETPROMPTWINDOW WINDOW))
  state filename directory host type rep xPos yPos penFlag fullFilename)

```

(\* \* check that all required data is specified)

```

(CLEARW promptW)
(SETQ state (MAPL.ExpandFilename ITEM WINDOW BUTTONS))
(SETQ filename (LISTGET state (QUOTE FILENAME)))
(COND
  ((EQUAL filename "")
    (PRIN1 "Unspecified file name." promptW)
    (RETURN)))
(SETQ directory (LISTGET state (QUOTE DIRECTORY)))
(COND
  ((EQUAL directory "")
    (PRIN1 "Unspecified directory." promptW)
    (RETURN)))
(SETQ host (LISTGET state (QUOTE HOST)))
(COND
  ((EQUAL host "")
    (PRIN1 "Unspecified host (DSK,IVY,etc.)" promptW)
    (RETURN)))

```

[SETQ xPos (MKATOM (LISTGET state (QUOTE XPOS)) (\* if xPos is not a number, set it to NIL for a Time-Series plot)

```

(COND
  ((NOT (NUMBERP xPos))
    (SETQ xPos (NUMBERP xPos))
    (PRIN1 "Time-series plot..." promptW)))
[SETQ yPos (MKATOM (LISTGET state (QUOTE YPOS))
(COND
  ((NOT (NUMBERP yPos))
    (PRIN1 "Unspecified position of 'Y' variable" promptW)
    (RETURN)))
(SETQ type (LISTGET state (QUOTE TYPE)))
(COND
  ((NOT type)
    (PRIN1 "Unspecified plot type (XY, etc)." promptW)
    (RETURN)))

```

(\* if xPos is NIL, do not allow Logs of the "time")

```

(COND
  ([AND (NOT xPos)
    (OR (EQUAL type (QUOTE LogX-Y))
      (EQUAL type (QUOTE LogX-LogY))
      (PRIN1 "Log of X (time) not permitted" promptW)
      (RETURN)))
(SETQ rep (LISTGET state (QUOTE REP)))
(COND
  ((NOT rep)
    (PRIN1 "Unspecified represent. (Points,etc.)" promptW)

```

```

(RETURN)))
[SETQ penFlag (EQ (QUOTE ON)
                  (LISTGET state (QUOTE PEN)

(* * make fullFilename)

(SETQ fullFilename (PACKFILENAME (QUOTE HOST)
                                host
                                (QUOTE DIRECTORY)
                                directory
                                (QUOTE BODY)
                                filename))

(* * if fullFilename is valid, then make plot)

(COND
  ((NOT (INFILEP fullFilename))
   (PRIN1 "File not found" promptW)
   (RETURN))
  (T (PRIN1 "Making plot ..." promptW)
      (MAPL.Gen.Plot fullFilename promptW xPos yPos rep type penFlag)
      (CLEARW promptW)
      (PRIN1 "Done" promptW)
      (RETURN]))

```

**(MAPL.Gen.Plot**

```
[LAMBDA (file promptW xPos yPos rep type penFlag) (* DSB "5-Dec-86 17:41")
```

```
(* * makes a plot from the general plot input format)
```

```
(PROG (dataList keyword plot objectList (pen 1)
      (plotNumber 1))
```

```
(* * get a data list of those pairs of data you want to plot)
```

```
(SETQ dataList (MAPL.Gen.ASCIIToLisp file xPos yPos)) (* (PRIN1 dataList PROMPTWINDOW))
```

```
(* * parse and plot)
```

```
[COND
  ((NOT dataList)
   (RETURN (PRIN1 "There is no data" promptW)
   (for item in dataList do (SETQ keyword (CAR item))
                        (SELECTQ keyword
                          (NEWPLOT (SETQ plot (MAPL.Gen.NewPlot plotNumber))
                                   (* New plot)
                          )
                          (START (SETQ objectList NIL)
                                   (* Start new object)
                          )
                          (END (SETQ objectList (MAPL.Gen.PlotObject plot objectList pen rep))
                                   (* Plot the object)
                          )
                          (ENDPLOT (SETQ plotNumber (MAPL.Gen.EndPlot plot plotNumber))
                                   (* End of plot)
                          )
                          (NEWPEN (SETQ pen (MAPL.Gen.NewPen item penFlag))
                                   (* New pen)
                          )
                          (SETQ objectList (MAPL.Gen.CollectData objectList item type))

```

**(MAPL.Gen.ASCIIToLisp**

```
[LAMBDA (file xPos yPos) (* DSB "5-Dec-86 17:34")
```

```
(* * returns ASCII data from a file in a list, with one sub-list per line.
The file must end in a CR.)
```

```
(PROG ((tempFile (QUOTE {core}tempplot.dat))
      (keywordList (QUOTE (NEWPLOT ENDPLOT START END NEWPEN)))
      (fileStream dataList lineList key shortList)
      (COND
        ((NOT (INFILEP file))
         (RETURN NIL))
        (COPYFILE file tempFile)

```

```
(* * copy to {core} because reads from filestream to {core} are much faster than reads from filestream to a VAX on the network.)
```

```
(* * if Keyword, return the line as is; otherwise, only include the numbers in the positions requested.)
```

```
(* * if xPos is NIL, put NIL in the first position of the short list)
```

```
(SETQ fileStream (OPENSTREAM tempFile (QUOTE INPUT)))
(while (NOT (EOFP tempFile)) do (SETQ lineList (MAPL.ReadASCIIline fileStream))
      (SETQ key (CAR lineList))

```

```

(COND
  ((MEMBER key keywordList)
   (SETQ shortList lineList))
  [(NOT xPos)
   (SETQ shortList (LIST NIL (CAR (NTH lineList yPos))
   (T (SETQ shortList (LIST (CAR (NTH lineList xPos))
   (CAR (NTH lineList yPos))
   (SETQ dataList (CONS shortList dataList)))

(CLOSEF fileStream)
(DELETEFILE tempFile)
(RETURN (REVERSE dataList])

```

**(MAPL.Gen.NewPlot**

```
[LAMBDA (plotNumber) (* DSB " 5-Dec-86 11:44")
```

```
(* * starts a new plot with Log scaling)
```

```

(PROG [newPlot (leftFirst 471)
  left
  (bottomFirst 50)
  bottom region [rightMenuItems (QUOTE ((Logscale SCAT.LOGSCALE "Toggle exponential tics"
    (SUBITEMS (X% axis (SCAT.LOGSCALE (QUOTE X))
    "X axis only")
    (Y% axis (SCAT.LOGSCALE (QUOTE Y))
    "Y axis only"))))
    (Coordinates SCAT.WORLDCOORD "Display world coordinates
    at cursor position"]
    (pointMenuItems (QUOTE ((Coordinates SCAT.POINTCOORDS "Display point coordinates"]
[SETQ left (PLUS leftFirst (TIMES 25 (DIFFERENCE plotNumber 1]
[SETQ bottom (PLUS bottomFirst (TIMES 25 (DIFFERENCE plotNumber 1]
[SETQ region (CREATEREGION left bottom 250 250))
[SETQ newPlot (CREATEPLOT NIL region "General Plot")]
(PLOTADDMENUITEMS newPlot (QUOTE RIGHT)
  rightMenuItems)
(PLOTMENUITEMS newPlot (QUOTE POINTMENU)
  (APPEND (PLOTMENUITEMS newPlot (QUOTE MIDDLE))
  pointMenuItems))
(RETURN newPlot])

```

**(MAPL.Gen.PlotObject**

```
[LAMBDA (plot objectList pen rep) (* DSB " 8-Dec-86 07:28")
```

```
(* * plots the objectList, with specified pen and according to the chosen representation
(e.g., points, curve, both))
```

```
(* * when plotting points, ignore the pen size)
```

```

(PROG NIL
[COND
  ((NOT plot)
   (RETURN (PRIN1 "ERROR: NEWPLOT command omitted. " PROMPTWINDOW])
[COND
  [(AND objectList (GREATERP (LENGTH objectList)
    0))
   (COND
    ((EQUAL rep (QUOTE Curve))
     (PLOTCURVE plot objectList NIL pen NIL T))
    ((EQUAL rep (QUOTE Points))
     (PLOTPOINTS plot objectList NIL NIL NIL T))
    ((EQUAL rep (QUOTE Both))
     (PLOTCURVE plot objectList NIL pen NIL T)
     (PLOTPOINTS plot objectList NIL NIL NIL T)
    (T (PROMPTPRINT "ERROR: PlotObject is NIL and not drawn"))))
(RETURN NIL])

```

**(MAPL.Gen.EndPlot**

```
[LAMBDA (plot plotNumber) (* DSB " 4-Dec-86 14:17")
```

```
(* * opens the plotwindow, and returns a new increment for the next plot position)
```

```

(PROG NIL
  (OPENPLOTWINDOW plot)
  (RETURN (PLUS plotNumber 1])

```

**(MAPL.Gen.NewPen**

```
[LAMBDA (item penFlag) (* DSB " 8-Dec-86 07:30")
```

```
(* * If penFlag is OFF, sets pen to 1; otherwise, sets pen width as instructed.)
```

```

(PROG (pen)
  (COND
    (penFlag (SETQ pen (CADR item)))
    (T (SETQ pen 1)))

```

```

70 "0000000000000000L@@" "0000000000000000L@@" "L@eeeeeeeeeeOn@ee@L@@" "L@eeeeeeeeG@ee@L@@"
"L@ee@C@000000000000L@@" "L@ee@B@ee@C@A@H@e@D@L@@" "L@ee@B@ee@ee@G@L@e@D@L@@" "L@ee@B@ee@ee@G@N@e@D@L@@"
"L@ee@B@ee@ee@C@e@D@L@@" "L@ee@B@ee@ee@C@A@H@e@D@L@@" "L@ee@B@ee@ee@C@O@H@e@D@L@@" "L@ee@B@ee@ee@C@G@L@e@D@L@@"
"L@ee@B@ee@ee@C@L@e@D@L@@" "L@O@H@B@ee@ee@C@A@N@e@D@L@@" "L@A@B@B@F@ee@ee@F@e@D@L@@" "L@A@H@B@ee@ee@C@e@D@L@@"
"L@ee@B@ee@ee@C@A@H@e@D@L@@" "L@ee@B@ee@C@L@e@G@H@e@D@L@@" "L@ee@B@F@ee@F@F@A@L@e@D@L@@" "L@A@O@B@B@e@L@C@e@D@L@@"
"L@O@I@O@B@A@H@A@N@e@D@L@@" "L@M@H@B@B@C@e@O@H@e@D@L@@" "L@M@H@B@F@e@G@e@e@e@D@L@@" "L@A@H@B@B@L@e@e@e@e@D@L@@"
"L@A@H@B@B@H@e@ee@ee@D@L@@" "L@A@H@B@B@A@H@e@ee@ee@D@L@@" "L@A@H@B@F@A@ee@ee@ee@D@L@@" "L@O@H@B@B@C@e@ee@ee@D@L@@"
"L@ee@B@B@B@ee@ee@D@L@@" "L@ee@B@B@F@ee@ee@ee@D@L@@" "L@ee@B@B@F@e@ee@ee@D@L@@" "L@ee@B@B@B@ee@ee@D@L@@"
"L@ee@B@B@ee@ee@ee@D@L@@" "L@ee@B@B@ee@ee@ee@D@L@@" "L@ee@B@B@O@O@O@O@N@D@L@@" "L@ee@B@B@B@B@B@B@B@D@L@@"
"L@ee@B@ee@ee@ee@ee@D@L@@" "L@ee@C@O@O@O@O@O@O@L@@" "L@ee@ee@ee@ee@ee@ee@L@@" "L@ee@ee@ee@ee@ee@ee@L@@"
"LO@O@O@H@e@ee@ee@ee@L@@" "L@LO@L@L@e@ee@ee@ee@L@@" "L@A@H@G@F@ee@ee@ee@ee@L@@" "L@C@C@C@e@ee@ee@ee@L@@"
"L@C@C@C@e@ee@ee@ee@L@@" "L@C@C@C@e@ee@ee@ee@L@@" "L@C@C@C@e@O@N@G@O@N@A@L@@" "L@C@C@C@e@N@F@N@G@F@A@H@@"
"L@C@C@C@e@L@B@L@C@F@A@H@@" "L@C@C@C@e@H@e@L@C@F@A@H@@" "L@C@C@C@e@L@H@L@C@F@A@H@@" "L@C@C@C@e@O@H@L@C@F@A@H@@"
"L@C@C@C@e@L@H@L@C@F@A@H@@" "L@C@G@H@C@e@H@e@L@C@F@A@H@@" "L@C@ee@C@e@L@B@L@C@F@A@H@@" "L@C@ee@C@e@N@F@L@C@G@H@@"
"L@ee@C@e@O@N@A@L@C@H@O@H@@" "L@ee@C@e@ee@ee@ee@ee@L@@" "L@O@H@e@G@L@ee@ee@ee@L@@" "L@O@H@e@G@L@ee@ee@ee@L@@"
"L@ee@ee@ee@ee@ee@ee@L@@" "L@ee@ee@ee@ee@ee@ee@L@@" "L@ee@ee@ee@ee@ee@ee@L@@" "L@ee@ee@ee@ee@ee@ee@L@@"

```

```
(RPAQQ BackgroundMenu NIL)

(PUTPROPS MATHSERVERPLOT COPYRIGHT ("Xerox Corporation" 1986))
```

---

FUNCTION INDEX

MAPL.ExpandFilename .....	2	MAPL.Gen.TopLevel .....	9	MAPL.Meta.Plot .....	3
MAPL.Gen.ASCIIToLisp .....	11	MAPL.MakeReadtable .....	1	MAPL.Meta.TopLevel .....	2
MAPL.Gen.CollectData .....	13	MAPL.Meta.ASCIIToLisp .....	4	MAPL.ReadASCIIline .....	2
MAPL.Gen.EndPlot .....	12	MAPL.Meta.DrawCom .....	4	MAPL.Simple.FreeMenu .....	6
MAPL.Gen.FreeMenu .....	9	MAPL.Meta.EndPlotCom .....	4	MAPL.Simple.MakeIconWindow .....	6
MAPL.Gen.MakeIconWindow .....	10	MAPL.Meta.FreeMenu .....	2	MAPL.Simple.MakePlot .....	6
MAPL.Gen.MakePlot .....	10	MAPL.Meta.MakeIconWindow .....	3	MAPL.Simple.Plot .....	7
MAPL.Gen.NewPen .....	12	MAPL.Meta.MakePlot .....	3	MAPL.Simple.TopLevel .....	5
MAPL.Gen.NewPlot .....	12	MAPL.Meta.MoveCom .....	4	MAPL.TopLevel .....	1
MAPL.Gen.Plot .....	11	MAPL.Meta.NewPenCom .....	5		
MAPL.Gen.PlotObject .....	12	MAPL.Meta.NewPlotCom .....	4		

---

VARIABLE INDEX

BackgroundMenu .....	14	MAPL.Gen.IconMask .....	14	MAPL.Simple.Icon .....	8
BackgroundMenuCommands .....	14	MAPL.Meta.Icon .....	5	MAPL.Simple.IconMask .....	9
MAPL.Gen.Icon .....	13	MAPL.Meta.IconMask .....	5	MATHSERVERPLOTOMS .....	1

---